AD-A128 871　　DESCRIPTION OF A DIGITAL SIMULATION PROGRAM FOR　　1/2
　　　　　　　　　MODELLING TACTICAL ANTI-R..(U) MASSACHUSETTS INST OF
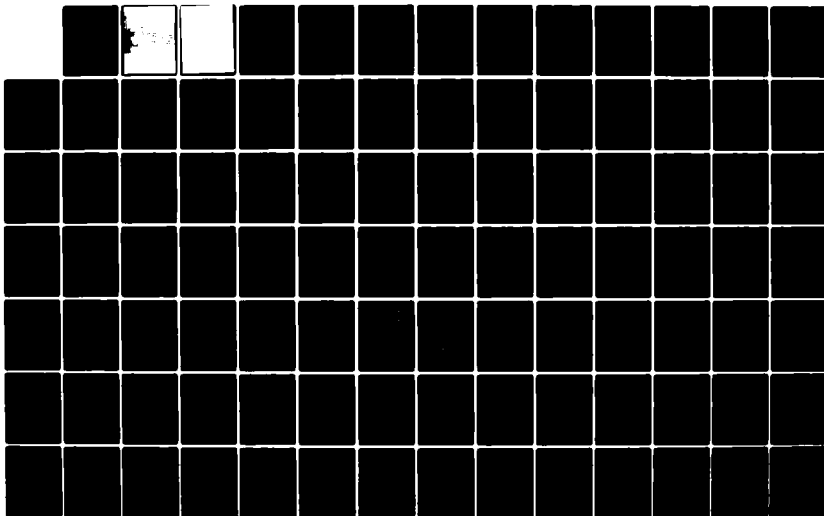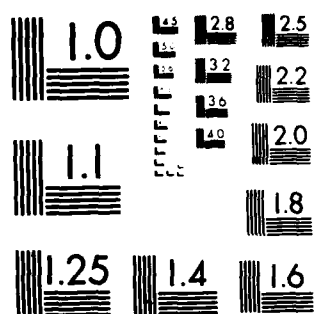　　　　　　　　　TECH LEXINGTON LINCOLN LAB　C M SORRENTINO ET AL.
UNCLASSIFIED　28 MAR 83 TST-69 ESD-TR-83-005　　　　　F/G 9/2　　　NL

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963 A

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

DESCRIPTION OF A DIGITAL SIMULATION
PROGRAM FOR MODELLING TACTICAL
ANTI-RADIATION WEAPONS
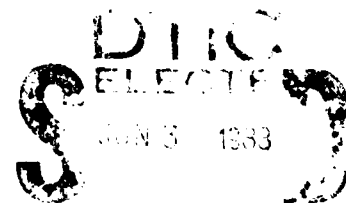
C.M. SORRENTINO

Group 44

R. D'AMATO

Consultant

PROJECT REPORT TST-69

(Tactical Systems and Technology)

28 MARCH 1983

Approved for public release; distribution unlimited.

LEXINGTON   **MASSACHUSETTS**

Abstract

A general description is given of the computer program which simulates the performance of an anti-radiation homing weapon taking into account the properties of the seeker, assumed to be body-referenced, and the air vehicle. For the purpose of illustration, sample calculations have assumed a minidrone as the air vehicle, although the simulation has a wide range of applications to a variety of platforms. The program can simulate terminal homing in the situations where multiple, co-channel CW emitters are deployed. The three constituent simulation blocks (signal in space, seeker, and tactical vehicle) are discussed in detail, as well as models for sensor errors and wind effects.

iii

CONTENTS

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION AND OVERVIEW

A computer program has been developed for the purpose of evaluation of an emitter homing seeker developed by Group 44. The details of the computations required to model the electromagnetic environment, and seeker and vehicle characteristics are given in various technical reports. In particular, the model for the electromagnetic environment is described in [1]. The simulation can be considered to consist of four parts:

(1) An initialization section

(2) A signal in space section, modelling target attributes and various multipath effects such as specular and diffuse reflections off the ground, obstacles and shadowing

(3) A seeker module, modelling antenna pattern, signal processing algorithm, and tracker and guidance logic

(4) A control and aerodynamics block, modelling vehicle autopilot and aerodynamics.

The driver for these parts is Program $JCMLST. For illustrative purposes, the vehicle and seeker that are considered are the XBQM-106 minidrone and the enhanced interferometer.

A generalized block diagram is given in Fig. 1.1. The overall simulation program sequence is illustrated in Fig. 1.2. This simulation sequence diagram is typical of the diagrams which will be used to describe the structure of the program. The blocks in the diagram represent programs, subprograms or functions, and the arrows indicate the direction of the call and return. As may also be seen on this diagram, an indentation in the blocks is used to emphasize the heirachy of calling dependence.

The simulation components are covered in Section 2.0 through 5.0. Section 6.0 discusses the EXEC files which are used to run the program on the Amdahl 480 computer at Lincoln Laboratory. Section 7.0 presents an example of a typical simulation output.

Appendices A through D give further details of the individual blocks illustrated in the program sequence diagrams, including input and output as well as definitions of the variables used. These appendices are most useful when

```
                 ┌─────────────────────────────────────┐
                 │         INITIALIZATIONS             │
                 │  READ IN INPUT DATA FROM EXEC       │
                 └─────────────────────────────────────┘
                                    │
                                    ●
                                    │
                                    ▼
                 ┌─────────────────────────────────────┐
                 │  DETERMINE MULTIPATH QUANTITIES     │
                 │  ASSOCIATED WITH                    │
                 │    (A)  DIRECT WAVE                 │
                 │    (B)  GROUND REFLECTIONS          │
                 │    (C)  BUILDING REFLECTIONS        │
                 │    (D)  AIRPLANE REFLECTIONS OFF OF │
                 │         AIRPLANES IN THE VICINITY   │
                 │         OF THE EMITTERS             │
                 └─────────────────────────────────────┘
                                    │
                                    ▼
                 ┌─────────────────────────────────────┐
                 │  DETERMINE PLANAR ANGLES OUTPUT     │
                 │  BY THE SEEKER BASED UPON           │
                 │  ANTENNA, SIGNAL PROCESSING,        │
                 │  AND TRACKER ALGORITHM              │
                 └─────────────────────────────────────┘
                                    │
                                    ▼
                 ┌─────────────────────────────────────┐
                 │  USE PLANAR ANGLES TO GENERATE      │
                 │  COMMANDS TO DRIVE VEHICLE TO       │
                 │  NEW POSITION IS SPACE              │
                 └─────────────────────────────────────┘
                                    │
                                    ●
                                    │
                                    ▼
                 ┌─────────────────────────────────────┐
                 │  GENERATE FINAL MISS DISTANCES      │
                 │  PLOT MULTIPATH, SEEKER, AND        │
                 │  VEHICLE PARAMETERS                 │
                 └─────────────────────────────────────┘
```
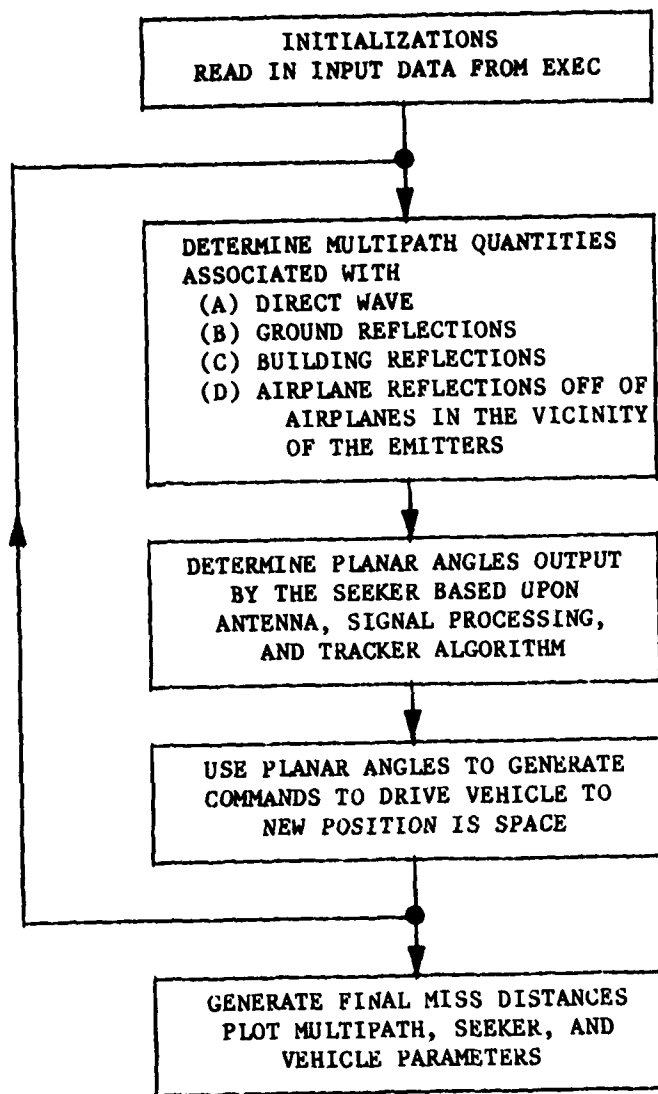
Fig. 1.1. Generalized simulation block diagram.
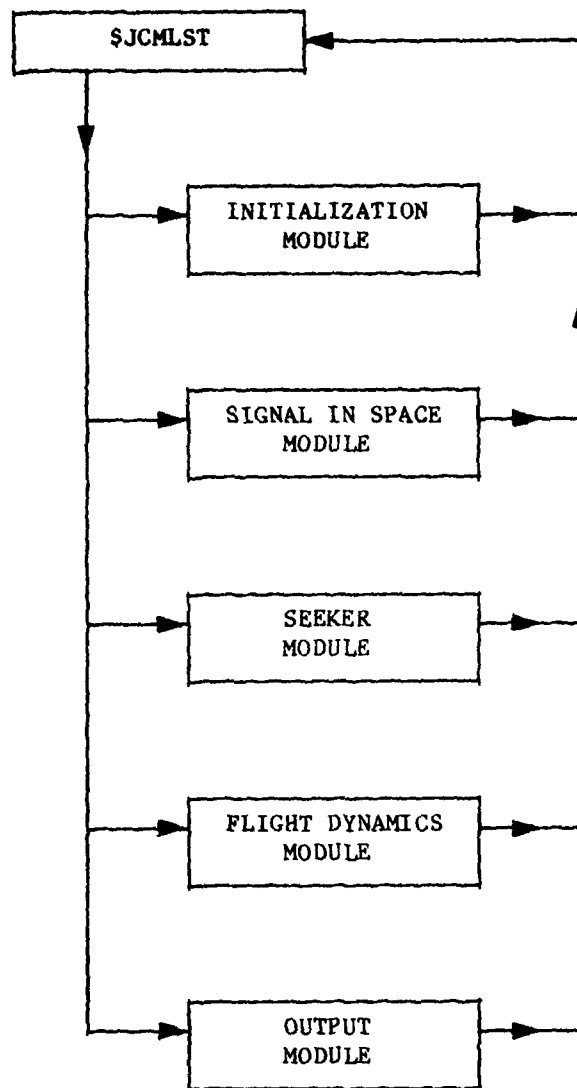
Fig. 1.2. Overall simulation program processing.

considered in conjunction with the program listings which accompany this report. The programs themselves are written primarily in FORTRAN and structured FORTRAN (IFTRAN).

2.0 INITIALIZATION MODULE

The program sequence for the initialization module is shown in Fig. 2.1. As noted in Section 1.0, the call to the module is from $JCMLST, the master driver of the simulation. The purpose of this module is to both initialize the appropriate variables as well as to read in information from the EXEC, which specifies the seeker angle processing options to be considered (such as the enhanced interferometer or amplitude monopulse). The input information specified by the user in block data file BDUHFM and the EXEC is written out to the terminal. Data specified in BDUHFM is manipulated to set up arrays which are used in the subroutines modelling the reflections off the ground, buildings, and aircraft. The driver $JCMLST considers each of the receiver options in turn and determines the vehicle trajectory and seeker performance for each of the direction finder algorithms considered. A number of criteria are employed to determine when to switch from consideration of one seeker to another. Following the initialization steps, multipath attributes, seeker calculations, and flight dynamics calculations are sequenced as shown in Fig. 1.2. This section will consider in a general way the flow of the initialization, while Appendix A will given additional details of the subroutines, along with definitions of the variables used in the program listings which accompany this report.

Subroutine PTIME is called initially to provide a beginning time which is subsequently used in calculations to determine the CPU time needed for multipath, seeker and flight dynamics procedures. MISINT is included for the initialization of two logical variables associated with the Maverick missile and is not applicable for the current simulation of the XBQM-106. The block labelled (1) indicates that several initializations are performed and emitter frequencies calculated with lines of code in the driver itself.

The subroutine READIT, which calls a number of other subroutines, reads the number which indexes the run and the title which labels the run. These are specified in the EXEC. The EXEC input is also used to determine whether specular ground reflections will be modelled as well as to provide initial conditions for the vehicle state vector (orientation, position, velocity).
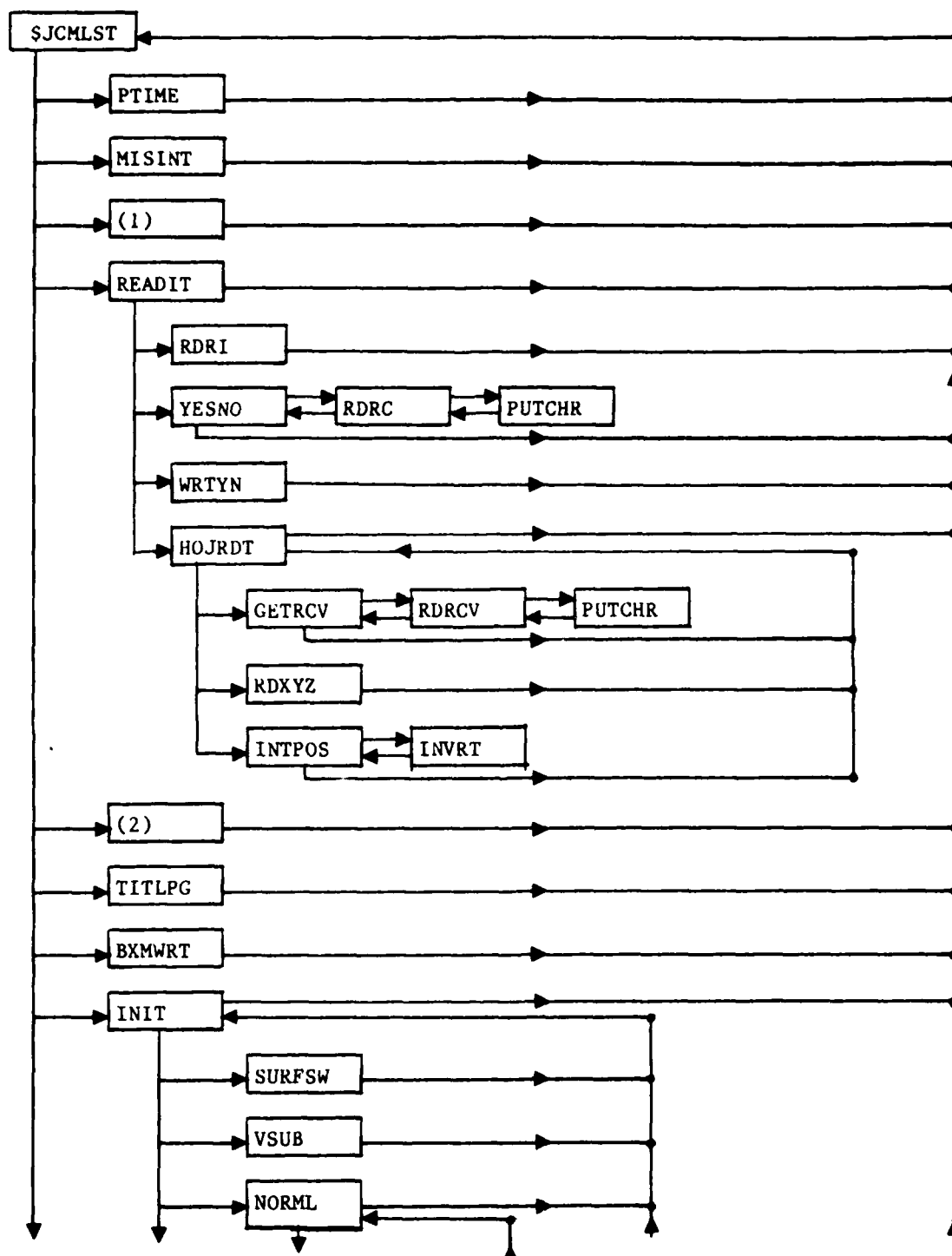
Fig. 2.1. Initialization program processing.

Fig. 2.1. (continued).

2-3
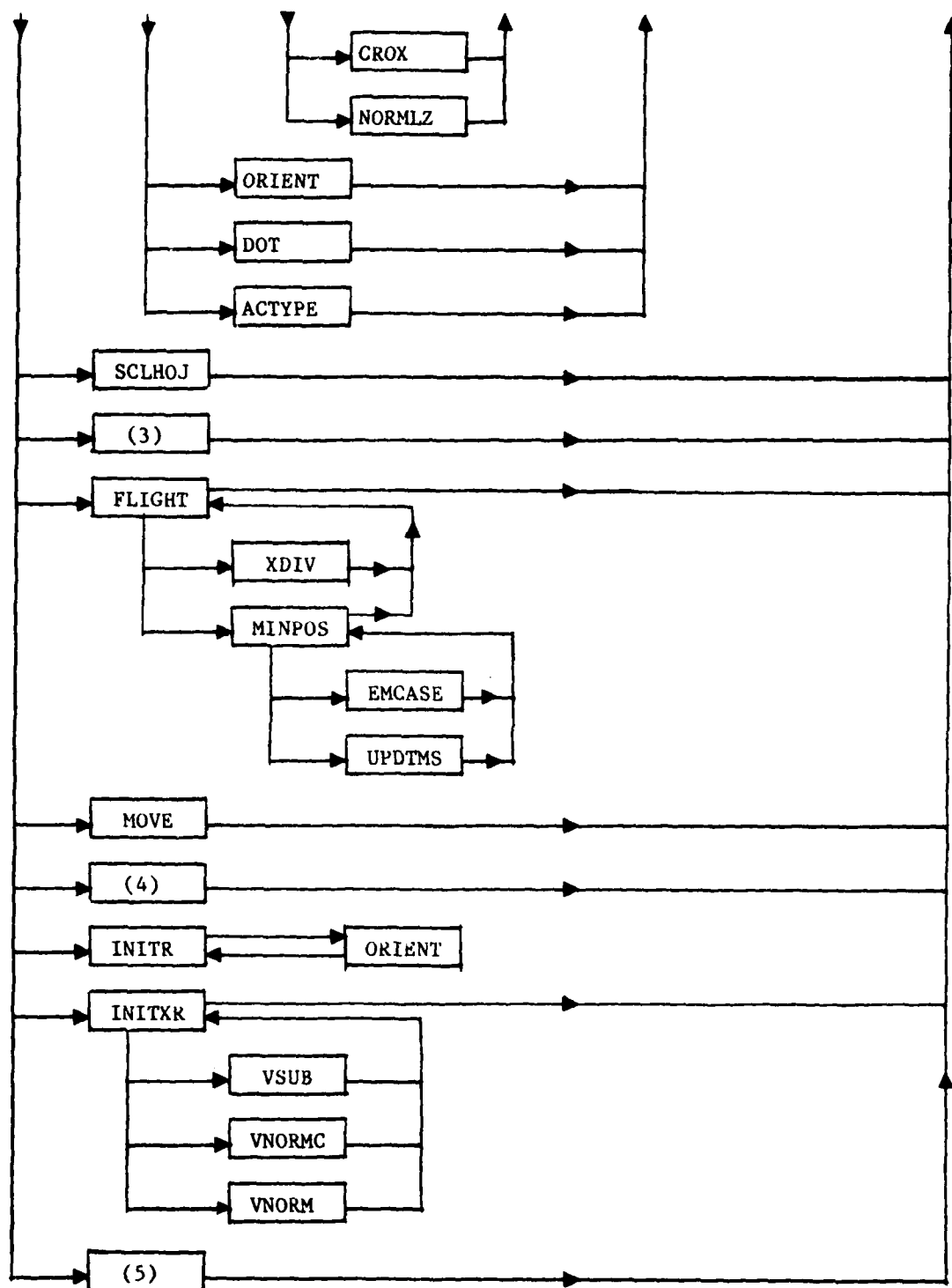
The subroutine RDRI reads the number used as a label to index the simulation run. The subroutine YESNO receives the user's response as to whether ground reflections are to be modelled. This subroutine calls RDRC which actually reads the input from the EXEC. The subroutine PUTCHR is used in the validation check of the user's response to the input questions. The subroutine WRTYN writes 'yes' or 'no' to the terminal regarding the modelling of specular ground reflections.

The subroutine HOJRDT reads which receivers are to be considered for the simulation run. Capability exists to model a wide variety of candidate antenna/receiver/signal processing systems for direction finding. It also accepts as input the vehicle state vector with respect to the reference inertial coordinate system. To accomplish these tasks a number of subroutines are invoked. GETRCV, RDRCV, AND PUTCHR perform the selection of the receiver option. RDXYZ reads the specified position of the vehicle in space while INTPOS, which calls INVERT, uses the initial specification of velocity and boresight orientation with respect to Emitter 1 to determine the vehicle boresight vector with respect to inertial space.

The block labelled (2) indicates that there are lines of code in the driver itself which checks the number of obstacles (buildings or aircraft), emitters and blinking information to insure that the number of input quantities remain within the allowed dimensions.

TITLPG writes out some of the parameters associated with the emitters, such as position, wave length, and polarization on the first page of the output. On the second page of the output, parameters associated with the multipath are written out.

BXMWRT writes out the terminal blinking emitter parameters, initial conditions of the vehicle, and parameters associated with the seeker.

The subroutine INIT, which, in turn calls a number of other subroutines, initializes the common areas for the signal in space routines based on the input data contained in the block data file BDUHFM.

2-4

The subroutine SURFSW modifies, if necessary, an array containing the corners of the rectangles and triangles describing the ground plane. Other subroutines that are subsequently called expect the corners to be entered in a certain order. This subroutine checks the appropriate array and, if necessary, modifies it to insure that the correct order is maintained.

The subroutines VSUB, ORIENT, DOT, and NORML along with its called routines CROX and NORMLZ are a series of vector manipulation subroutines needed for INIT. The subroutine ACTYPE specifies certain parameters for aircraft. For the simulations of scenarios which do not contain reflections off aircraft this subroutine has no effect on the results.

The call to the subroutine SCLHOJ establishes scaling parameters for the plotting programs.

The block labelled (3) represents lines of code in the driver $JCMLST for several initializations after which calls are made to the other main modules of the simulation to determine the time history of the seeker/vehicle system in the tactical scenario specified in the block data file BDUHFM.

Return to the driver is cycled with calls to the subroutine FLIGHT to determine when a given receiver loop is to be ended and final miss distances are to be determined. The criteria used for this are as follows:

(a) the altitude of the vehicle is less than 0.1 ft.

(b) the altitude of the vehicle is less than the height of a particular emitter and the distance from the emitter is less than 50 ft.

(c) the index which specifies the flight time exceeds a specified value.

The subroutines used in FLIGHT to accomplish these tasks are:

XDIV   performs the normalization of a three component vector

MINPOS   determines the miss distance to emitter IXMTR and calls

EMCASE   which determines the coordinates of a vector from the vehicle to the emitter.

UPFTMS   which determines the miss distance using a point of
         closest approach analysis.

The subroutine MOVE is used to load emitter parameters into a labelled
common area.

The block labelled (4) indicates lines in the driver used to make a num-
ber of initializations after which the subroutine INITR is called to compute
the receiver position and velocity coordinates in a reference frame of equiva-
lent vertical aircraft fuselages.

The subroutine INITXR computes the relative vector from the transmitter
to the receiver, the wave number and the wave number squared. This subroutine
calls the vector subroutines VSUB, VNORM and VNORM.

The final block designated (5) represents lines of code in the driver to
calculate the ratio of the distance from the receiver to Emitter 1 along with
the associated time delay.

3.0 SIGNAL IN SPACE MODULE

The signal in space module characterizes the electromagnetic environment as seen by the receive antenna. As shown in Fig. 3.1 this is done in a multi-emitter scenario by considering each of the emitters in turn and computing the multipath parameters associated with the direct wave from emitter to the receiver in the vehicle along with the specular reflections off the ground and other obstructions such as buildings and other aircraft in the vicinity of the emitters. It is to be noted that different bounce paths are assumed for reflections off the various obstructions. Figure 3.2, which describes the overall program flow of the module, shows that each of the major submodules, or subroutines, are called by the master simulation driver, $JCMLST.

Subsection 3.1 discusses the sequencing of the subroutines which are used in the direct wave computation, while subsections 3.2, 3.3, and 3.4 review the subroutine used for the specular ground, building, and airplane reflections, respectively. Subsection 3.5 discusses the parameters required for multipath computations. Appendix B provides additional details of the input and output, as well as the definition of variables given in the program listings which accompany this report.

In most of the simulation runs done for the EHT studies, building and aircraft reflections as well as shadowing due to ground humps were not modelled but are included in this description for completeness.

3.1 Direct Wave Computation

The program sequence for the direct wave computation is shown in Fig. 3.3. This part of the signal in space program determines the multipath parameters associated with the direct wave from the emitters including any shadowing effects associated with obstructions. The first step in the procedure is to invoke the subroutine CHKMLT which checks the number of multipath components, and, if there are more than a specified number, stops execution of the program. Next, the subroutine DIRWV computes the multipath parameters associated with the direct wave, including:

ENTER MULTIPATH COMPUTATION PROCEDURE

EXCEED EMITTER COUNT — YES

NO

DIRECT WAVE COMPUTATIONS

SPECULAR WAVE COMPUTATIONS

EXCEED BUILDING COUNT — YES

NO

EXCEED BOUNCE PATH COUNT — YES

NO

BUILDING REFLECTIONS

EXCEED AIRCRAFT COUNT — YES

NO

EXCEED BOUNCE PATH COUNT — YES

NO

FUSELAGE REFLECTIONS

TAILFIN REFLECTIONS

EXIT MULTIPATH COMPUTATION PROCEDURE

Fig. 3.1. Multipath computation procedure

3-2

```
                 ┌──────────┐ ◄─────────────────────────────────────┐
                 │ $JCMLST  │                                        │
                 └──────────┘                                        │
                      │                                              │
                      ▼                                              │
                 ┌──────────────────┐                               │
                 │  DIRECT WAVE     │                               │
            ────►│  COMPUTATION     ├────►─────────────●            │
                 │  SECTION         │                               │
                 └──────────────────┘                               │
                                                                    │
                 ┌──────────────────┐                               ▲
                 │ GROUND REFLECTION│                               │
            ────►│  COMPUTATION     ├────►─────────────●            │
                 │  SECTION         │                               │
                 └──────────────────┘                               │
                                                                    │
                 ┌──────────────────┐                               │
                 │BUILDING REFLECTION│                              │
            ────►│  COMPUTATION     ├────►─────────────●            │
                 │  SECTION         │                               │
                 └──────────────────┘                               │
                                                                    │
                 ┌──────────────────┐                               │
                 │AIRCRAFT REFLECTION│                              │
            ────►│  COMPUTATION     ├────►                          │
                 │  SECTION         │                               │
                 └──────────────────┘                               │
```
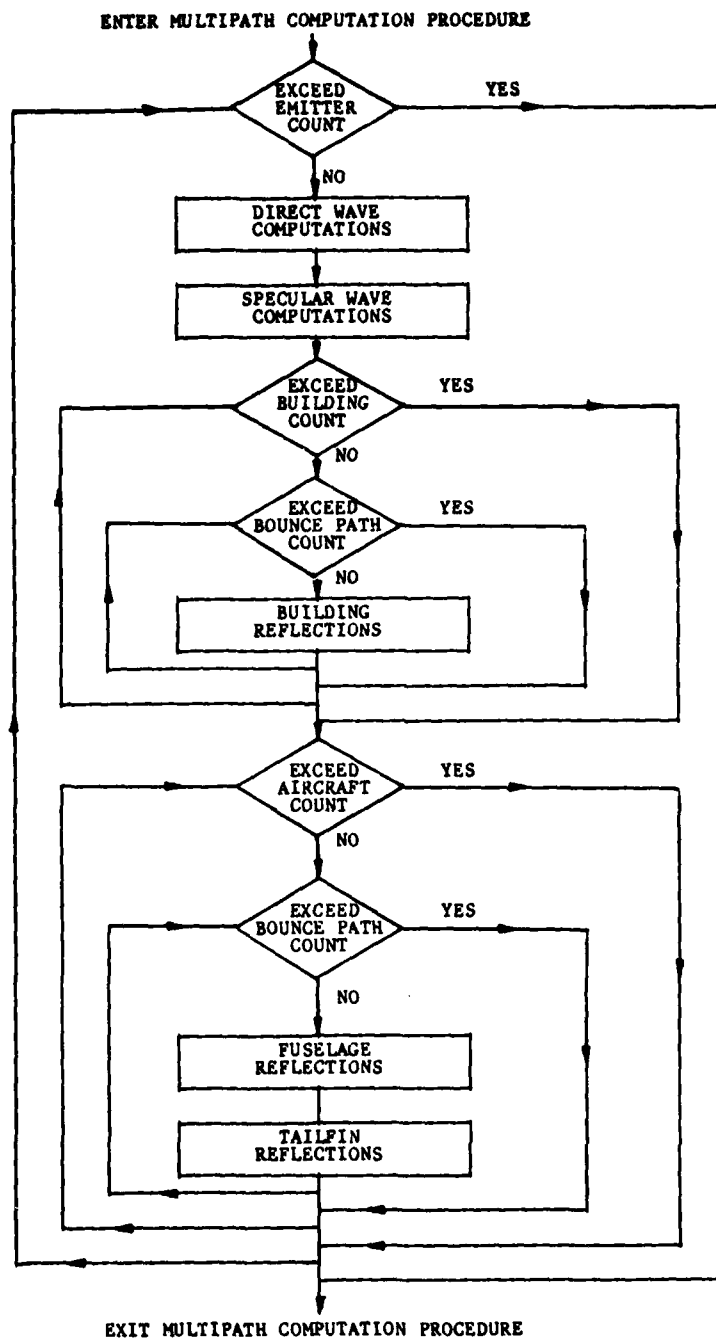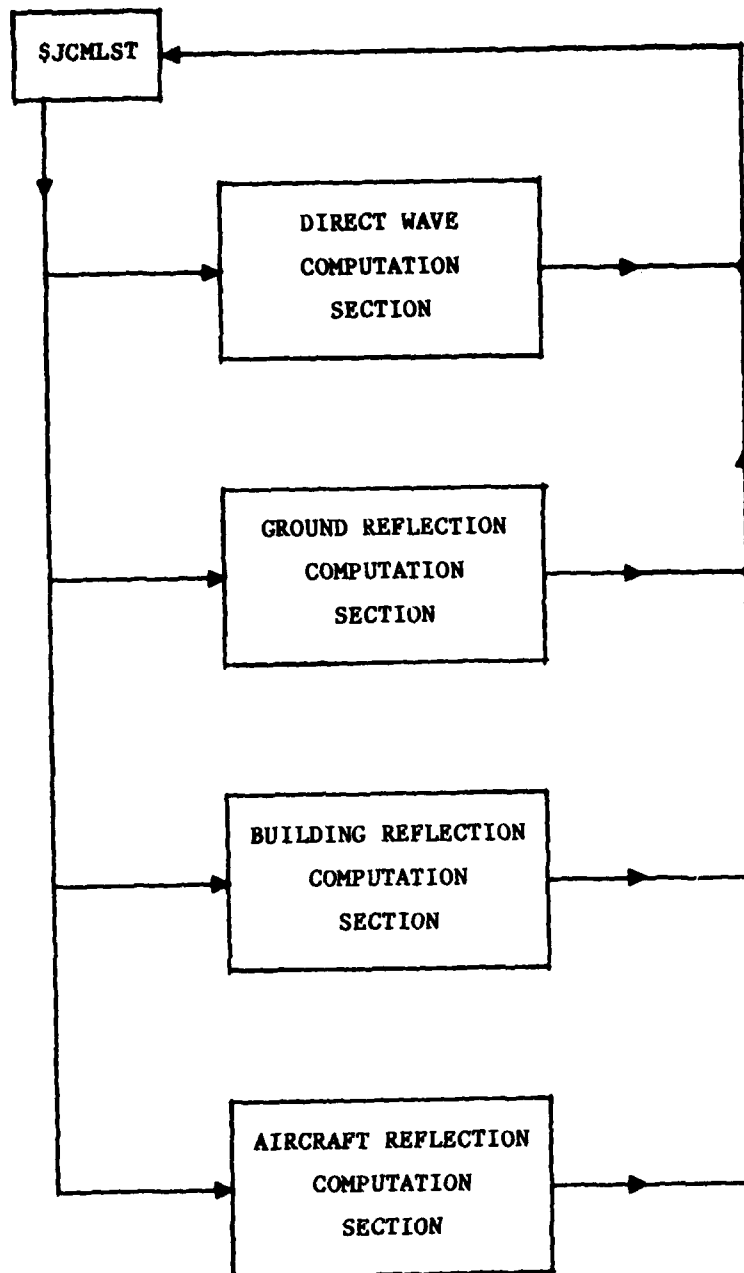
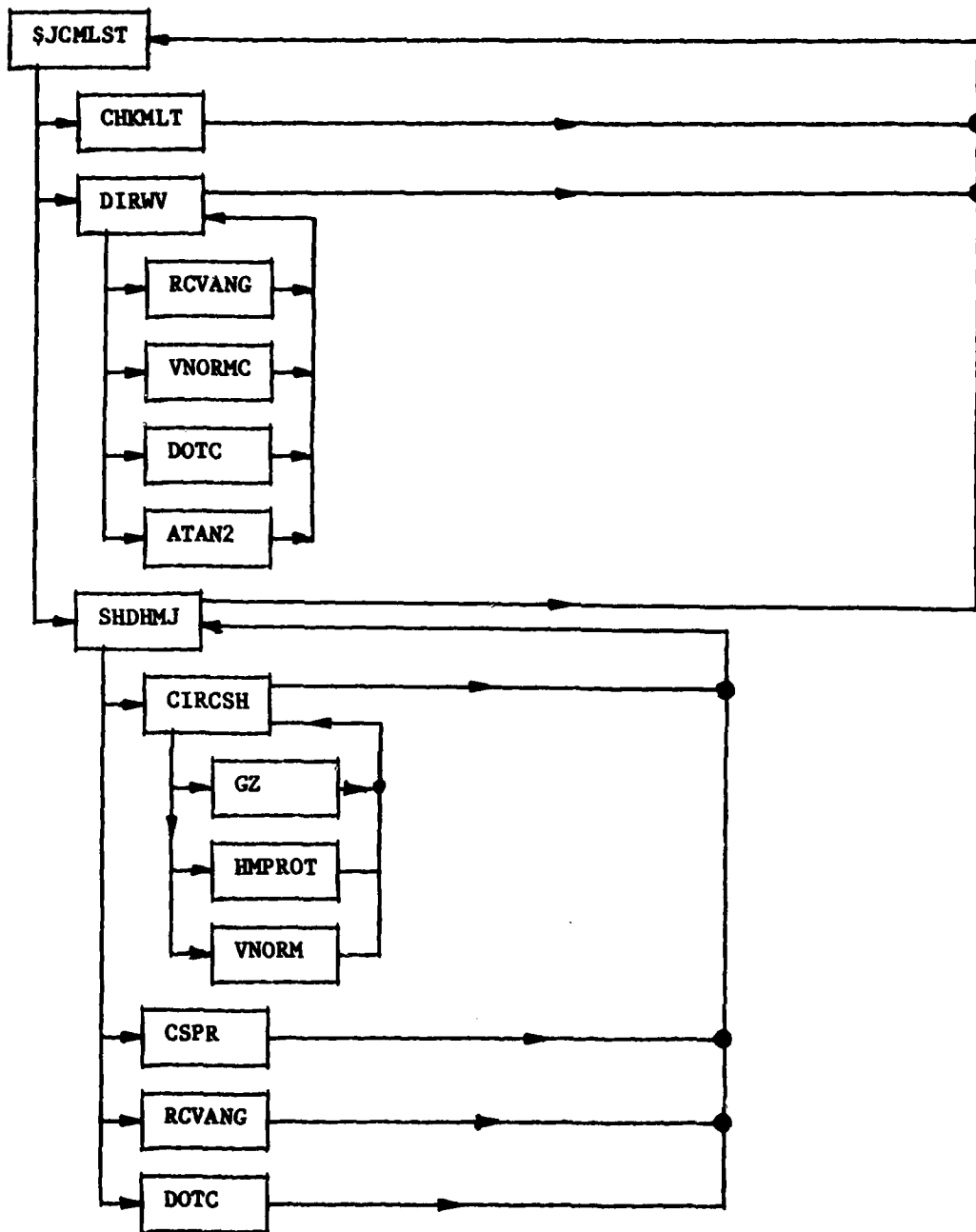Fig. 3.2 Overview of program processing for wave in space.

Fig. 3.3 Program processing for direct wave computations.

- The Doppler shift

- The planar angles which specify the direction of the incident radiation with respect to the receive antenna

- The planar angles which specify to the location of the receiver relative to the transmitter

In accomplishing these tasks, several subroutines are called:

RCVANG   which calculates planar angles in the receiver reference system given the coordinates relative to the reference inertial coordinates.

VNORMC   which computes the length of a three-component vector

DOTC   which determines the dot product of two vectors

ATAN   which finds the arc tangent

Next, the subroutine SHDHMJ models the shadowing effect due to circular cylindrical humps and determines the multipath parameters associated with these effects. Figure 3.4 shows the standard coordinate system used in the computations done in this subroutine. As noted in Fig. 3.3, subroutine SHDHMJ calls a number of other subroutines, the major two being CIRCSH and CSPR. Subroutine CIRCSH computes the parameters needed to determine the shadowing effect due to a circular cylindrical hump. CIRCSH calls, in turn, GZ which computes the complex function required to determine the shadowing of the hump and HMPROT which transforms inertially reference coordinates of a vector into a coordinate system based on the vector from the transmitter to the receiver. The subroutine CSPR determines the Fresnel integral.

The subroutine SHDHMJ will not be called if the parameter IHUMP is set to zero (its default value) in the block data file BDUHFM.

3.2 Ground Reflection Computation

Figure 3.5 outlines the sequence for ground reflection computation. As for the direct wave calculation, the subroutine CHKMLT is first called to insure that the number of multipath components does not exceed a specified
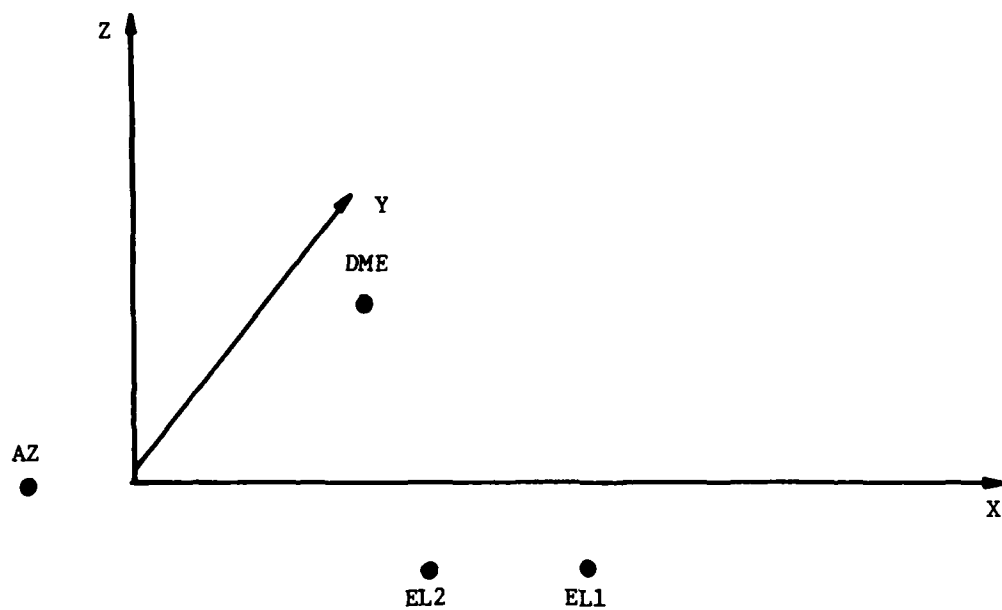
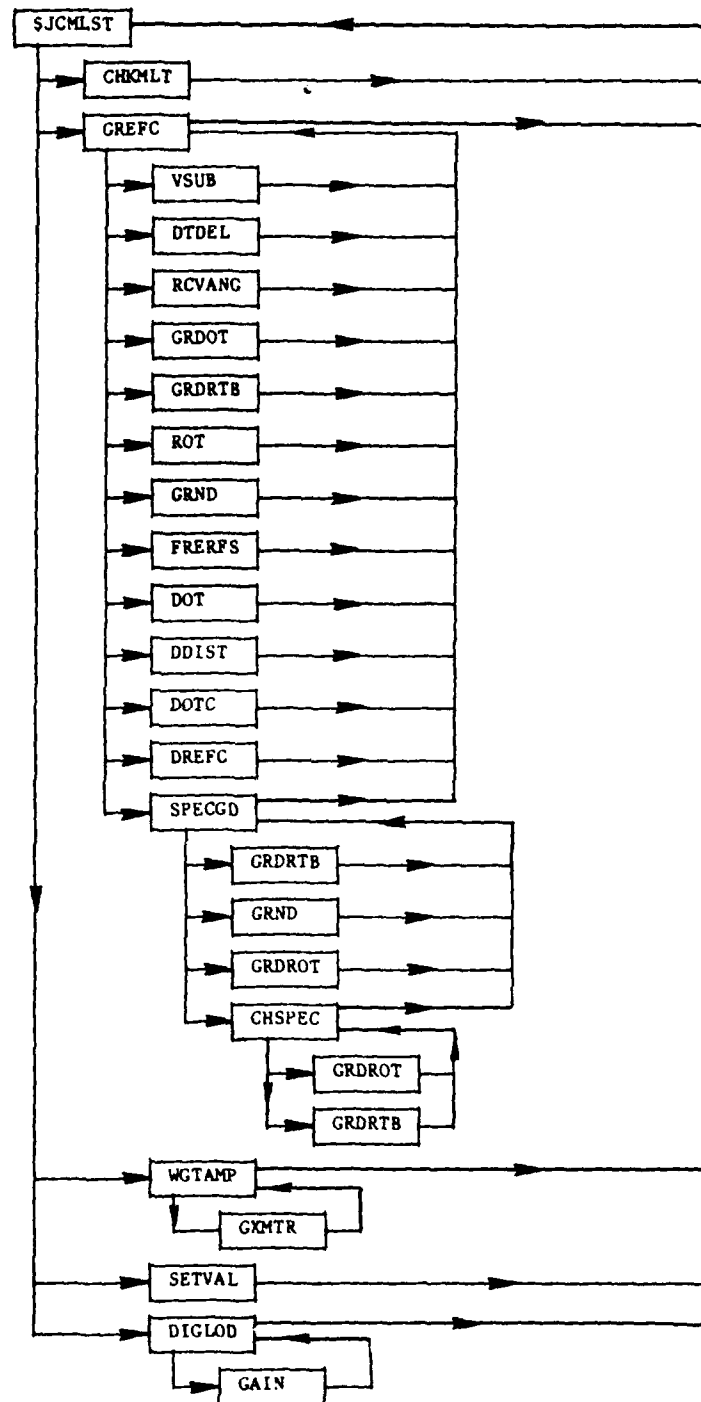Fig. 3.4. Reference coordinate system.

Fig. 3.5. Program processing for ground reflection computation

value. If specular ground reflections are modelled (as in most simulation cases), subroutine GREFC is then called. If not, then the parameters associated with multipath are set to zero. If the parameter ISPGRD is specified in block data BDUHFM as 1 (again, its default value) a simplified method is used to find the multipath parameters in subroutine FRERFS. On the other hand, if ISPGRD is zero, the Fresnel integral is calculated using a two-dimensional approximation. In the latter case, the ground plane can be represented by a set of arbitrarily oriented rectangles and triangles, each with its own characteristic surface roughness coefficient and dielectric constant. A flow chart describing the computation of multipath parameters is given in Fig. 3.6. As may be noted in this diagram specular reflection parameters are computed for each region or element. The simplified method bypass (for ISPGRD = 1) which uses default values of the surface roughness and dielectric constant may also be seen in the diagram.

Multipath parameters are adjusted by modelling the transmit antenna pattern in subroutines WGTAMP and GXMTR. The multipath parameters are loaded into labelled common areas MULTI and MULTIP by means of subroutine SETVAL. Subroutine DIGLOD transfers the parameters associated with the maximum amplitude multipath component into labelled common area DIAGP after being modified by the receiver antenna pattern which is characterized in GAIN.

Appendix B supplies additional information on the subroutine discussed and shown in Fig. 3.5.

3.3 Building Reflection Computations

The sequence of computation for building reflections is shown in Fig. 3.7. When buildings are included in a scenario, calculations are made for each building specified considering four different bounce paths as follows:

* transmitter-ground-building-receiver
* transmitter-building-ground-receiver
* transmitter-ground-building-ground-receiver
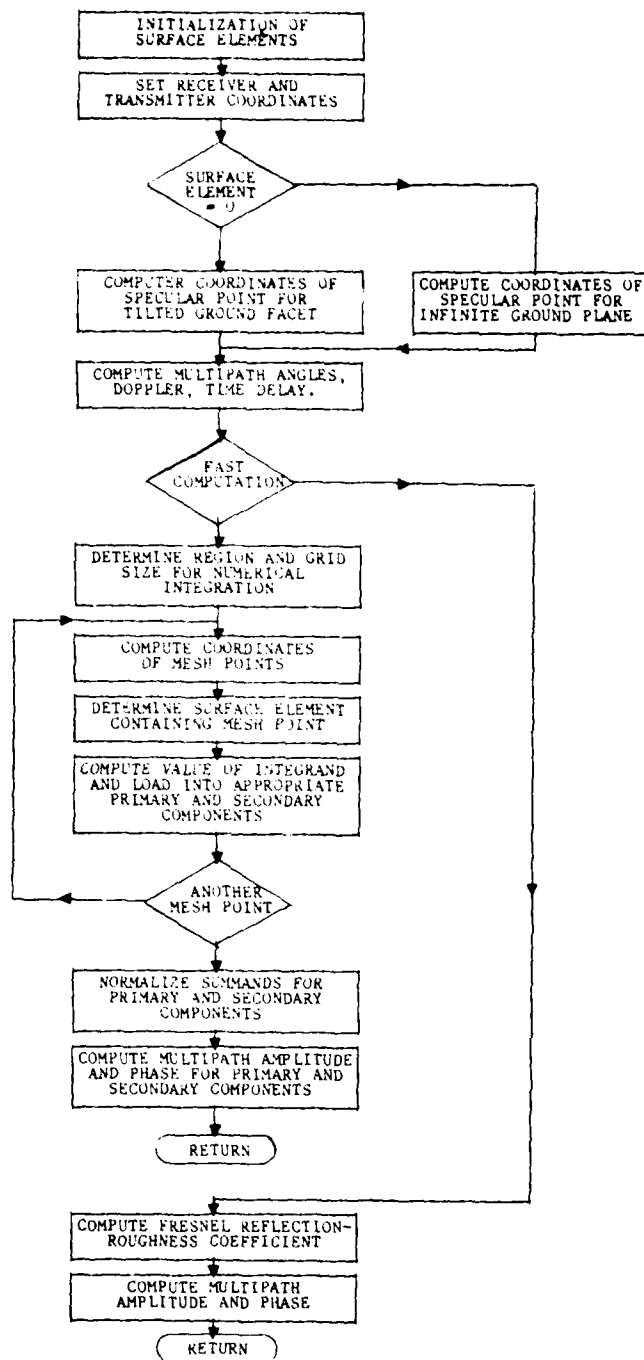* transmitter-building-receiver

Fig. 3.6. Flow chart for computation of multipath
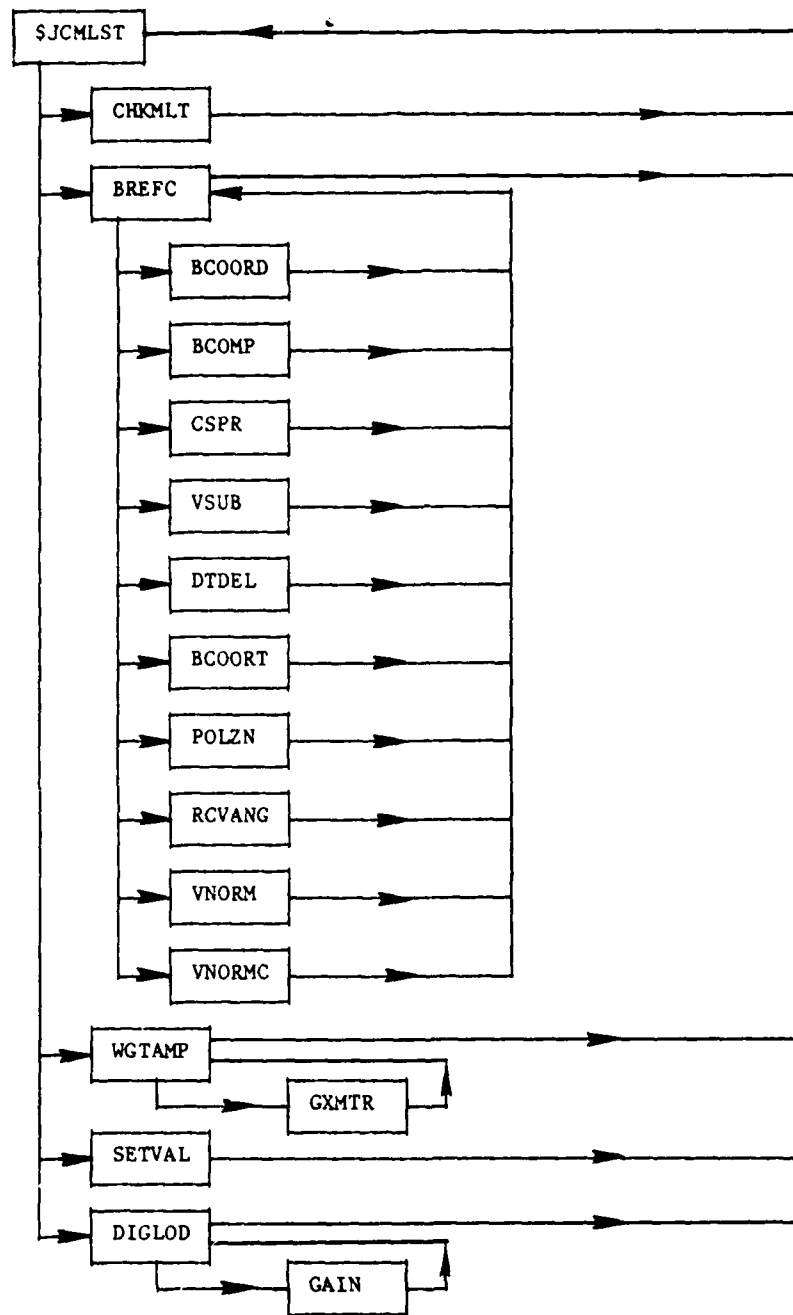parameters for specular ground reflection.

3-9

Fig. 3.7. Program processing for building reflections.

As for the other computation segments in this module, CHKMLT is called first to insure that the number of multipaths does not exceed the specified number. Next, the subroutine BREFC is called to calculate the parameters associated with reflection off buildings such as amplitude, phase, time delay, and Doppler shift. BREFC, in turn, calls a series of other subroutines as shown in Fig. 3.7. Subroutines BCOMP, BCOORD, BCOORT are coordinate transformation subroutines and VSUB, VNORM, VNORMC are vector operation subroutines. RCVANG is a subroutine that is used to calculate planar angles relative to vehicle axes given an inertially referenced vector from the vehicle to a point in space. The subroutine CSPR, discussed in Section 3.1, determines the Fresnel integral, while POLZN computes the equivalent Fresn l reflection coefficient for reflection from a tilted planar surface.

Subroutine WGTAMP modifies the specular reflection parameters by incorporating the effects of the transmitting antenna and uses the subroutine GXMTR which provides the model for the transmitting antenna. SETVAL transfers the multipath quantities to the labelled common areas MULTI and MULTIP. Finally, the subroutine DIGLOD transfers the maximum amplitude building multipath to the labelled common area DIAGP after it has been weighted with the receive antenna pattern modelled by subroutine GAIN.

It is to be noted that for the tactical scenarios associated with the EHT studies, the number of buildings specified in BDUHFM was normally set to zero so that building reflection segment was not invoked.

Appendix B supplies additional information on the subroutines shown in Fig. 3.7 and discussed briefly above.

3.4 Aircraft Reflection Computation

Figure 3.8 provides the program sequence for the computation of aircraft reflections. Comparing with Fig. 3.7 a general similarity in the program flow is noted. As for the case of buildings, four different bounce paths are considered are also considered for aircraft. For the aircraft case, however, separate consideration is given for fuselage components and vertical tail components. Both are modelled as cylinders: the fuselage as a horizontal cylinder and the tail as a vertical cylinder and the same basic programs are used.

3-11
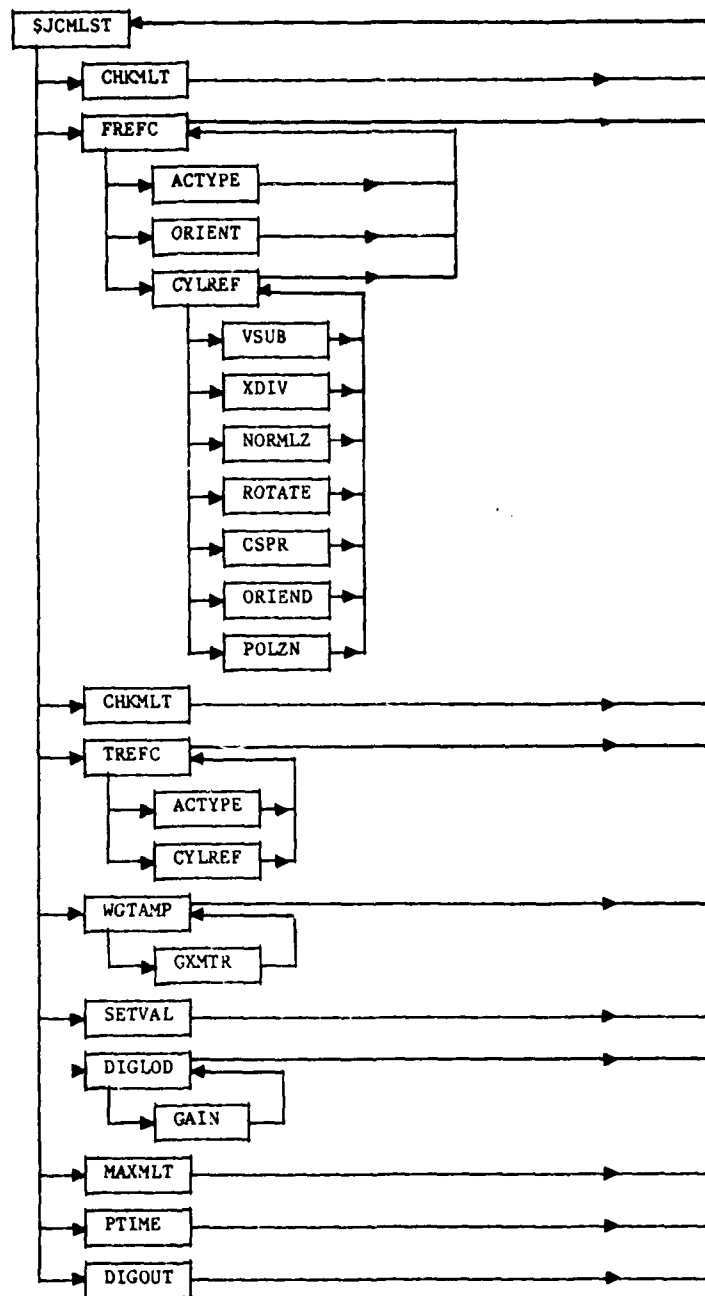
Fig. 3.8. Program processing for aircraft reflections.

3-12

Referring to Fig. 3.8, the subroutine CHKMLT is used twice to establish that the number of multipath components does not exceed a specified value. The first time is for the fuselage components and the second time is for the fin components. After the first call, the subroutine FREFC is called to make the calculation for the fuselage components. FREFC, in turn, calls a number of other subroutines:

ACTYPE determines the geometrical parameters associated with each aircraft

CYLREF computes the multipath parameters of a vertical cylindrical surface

ORIENT is used to compute the equivalent vertical cylinder so that CYLREF can be used for horizontal cylinders.

The subroutine CYLREF calls, in turn, a number of other subroutines:

VSUB, XDIV these subroutines are vector manipulation subroutines
NORMLZ, ROTATE
ORIEND

CSPR as noted previously, is used to compute the Fresnel integral.

POLZN computes the equivalent Fresnel reflection coefficients for reflection from a tilted planar surface.

After the second call of CHKMLT, the tail multipath contributions are computed using TREFC. Since the fins are vertical the subroutine ORIENT is not called, and the computations controlled by TREFC for the tail are similar to those for the fuselage.

As for the building computations, there are calls to WGTAMP, SETVAL, and DIGLOD. The subroutine WGTAMP modifies the specular reflection parameters by including the effects of the transmitting antenna and uses the subroutine

GXMTR to provide the model of the antenna pattern. SETVAL transfers the multipath quantities to the labelled common areas MULTI and MULTIP. The subroutine DIGLOD writes the maximum amplitude building multipath to the labelled common area DIAGP after it has been weighted with the receive antenna pattern modelled by subroutine GAIN.

The subroutine MAXMLT is used to determine the scaling parameters for multipath diagnostics at a particular evaluation point. A call to PTIME provides the time marker needed to compute the elapsed time employed by the total time needed to compute the multipath.

Finally, the subroutine DIGOUT writes the multipath data to the disk.

As for the case of building reflections, tactical scenarios associated with EHT studies, did not normally invoke reflections from aircraft so the number of aircraft in BDUHFM was set to zero.

Appendix B supplies additional data on the subroutines shown in Fig. 3.8.

3.5 Parameters Required for Multipath Computations

The following parameters specify the scenario which is used in the modelling of the signal in space seen by the EHT seeker. A standard rectangular coordinate system is used. All lengths, frequencies, and times are given in feet, Hz, and seconds, respectively. The symbols used are consistent with those given in [1]. In most cases, parameters are set to their default values so as to decrease the complexity of the multipath computations.

Transmitter Parameters (Azimuth, DME, Elevation 1, and Elevation 2)

1. X,Y,Z-coordinates of location of transmitter. XMTRAZ
2. Wavelength (nominally 0.2 feet for C-band). WLAZ
3. Polarization (vertical or horizontal). PLZAZ where
    TRUE implies vertical polarization
4. Dimension of transmitter antenna. DEFAULT = 0 (except
    for DIMDME which must be specified as nonzero)

3-14

5. Number of intervals when transmitter is off, which is less than or equal to 10 NAZBLK. Start and end time of blinking interval in seconds. TAZBLK (,)

6. Number of transmitters. NXMTR

7. Relative power levels of emitters. BXMPWR

8. Tracking option parameter for each transmitter. KXMTRK ( )

9. Planar azimuth pointing angle for each transmitter for tracking option 2. BAZTRK ( )

10. Planar elevation pointing angle for each transmitter for tracking option 2. BELTRK ( )

Specular Ground Reflection in Labelled Common SURFAC

1. Number of rectangular surface elements, which must be less than or equal to 10. NR

2. X,Y,Z-coordinate of two corners, plus X,Z-coordinates of a third corner, in increasing order of magnitude for the X-coordinate, for each rectangular surface element. SURFS (,)

3. The real and imaginary relative dielectric constants, denoted by ε_R, $-\varepsilon_I$, respectively, and the root-mean-square roughness height, σ_h, for each rectangular surface element. ERS (,) SH2S ( )

4. An indicator parameter, for each rectangular surface element, which is equal to 0 or 1 depending on whether its contribution to the specular ground reflection is to be considered a primary, or secondary, component, respectively [1]. The total number of such components is specified to be less than or equal to ten. NRSPEC

5. Number of triangular surface elements, which is less than or equal to 10. NT

6. X,Y,Z-coordinates of the three corners of each triangular surface element, in increasing order of magnitude for the X-coordinate. SURFS (,)

7. ϵ_R, $-\epsilon_I$, σ_h, for each triangular surface element. ERS, SH2S

8. An indicator parameter, for each triangular surface element, which is equal to 0 or 1 depending on whether its contribution to the specular ground reflection is to be considered a primary, or secondary, component, respectively [1]. NTSPEC

9. Default values of ϵ_R, $-\epsilon_I$, σ_h, which are used in those regions not specified by the previously defined rectangular and triangular areas. ERO, SH2O

10. Number of Fresnel zones used in integration region, nominally equal to 2.8. NFZ

11. Number of grid points used along the major and minor axes of the integration regions, which are equal to 25 and 11, respectively. NA and NB

12. A parameter which is 0 if the full numerical calculation of the Fresnel integral is to be calculated, and is 1 if the simplified Fresnel reflection-roughness coefficient is to be computed using FRERFS. ISPGRD

Scattering from Buildings in Labelled Common BUILD BDGRND

1. Number of buildings which should be less than or equal to 10. NBLD

2. Height of bottom edge of front face of building above ground, H_{bottom}, for each building. HBOT (   )

3. Height of building, relative to bottom edge, H_B, for each building. HB(   )

4. X,Y-coordinates of left-hand, and right-hand, edge of front face of building, for each building. BLD (,   )

5. ϵ_R, $-\epsilon_I$, σ_h for each building. DICSTB(), SH2B( )

6. ϵ_R, $-\epsilon_I$, σ_h for ground reflection.

7. Tilt angle of building. This angle is positive if building is above, and tilts away from, the centerline, or if building is below, and tilts toward, the center-line. TILT (   )

8. Differential height factor of ground (the height of ground). This factor is positive if the ground is above zero height level, and negative otherwise. GRNDBD (   )

Scattering From Aircraft ACS

1. Number of aircraft, which should be less than or equal to 10. NAC

2. Aircraft type, for each aircraft, e.g., 1(= 747), 2(=707-320B), 3(= 727), 4(= DC10), 5(= C-124), 6(= Convair 880), 7(= Hastings). A subroutine ACTYPE is called using the appropriate aircraft type. This subroutine is contained in program JCHOJ. It loads the following aircraft related quantities into storage. NACTYP ( )

2.1 Area of both wings. WA

2.2 Radius of fuselage. FR

2.3 Length of fuselage. FL

2.4 Radius of curvature of tail fin. TR

2.5 Width of tail fin. TL

2.6 Height of tail fin. TH

2.7 Height of center of fuselage above the ground. HF

2.8 Length of one wing. WL

2.9 Width of wing. WW

2.10 Thickness of wing. WT

Quantities 2.8 through 2.9 are found in labelled common WINGS.

3. X,Y-coordinates of cockpit and tail fin edge of fuselage centerline, for each aircraft. AC (,)AC

4. Altitude of aircraft for each aircraft, defined as the height of fuselage centerline above the ground.

5. Differential height factor of the ground.

Shadowing Due to Runway Hump in LABELLED COMMON HMPR

1. X,Y,Z-coordinates of front of hump. HUMPF

2. X,Y,Z-coordinates of center of hump. HUMP

3. X,Y,Z-coordinates of back of hump. HUMPB

4. Parameter which is zero if runway hump shadowing is not
    to be done and one otherwise. IHUMP

The following information is also specified:

1. Length of runway. RUNLEN

2. Width of runway. RUNWID

Default Specification of Multipath Quantities

1.	DIMDME	Nonzero	
2.	ISPGRD	1	compute specular reflection coefficients using fast computation.
3.	NR	0	no rectangles
4.	NT	0	no triangles
5.	NB	0	no buildings
6.	NAC	0	no aircraft
7.	IHUMP	0	no humps and shadowing effects

4.0 SEEKER MODULE

This section will discuss in a general way the program flow of the seeker module as was done in Sections 2.0 an 3.0 for the initialization and signal in space modules, respectively. Three main elements make up the seeker module: (1) the antenna model, (2) the signal processing algorithm and (3) the tracker algorithm. In addition to the description of the program flow, subsections 4.2 and 4.3 will provide some additional insight into the signal processing and tracker, respectively. More detail on the various subroutines will be given in Appendix C, along with definitions of the variables used in the program listings which accompany this report.

4.1 Program Sequence for Seeker Module

The program sequence for the seeker module is shown in Fig. 4.1. As previously noted, the call to this module is from the master simulation driver $JCMLST. The seeker module models the received antenna characteristics, signal processing and angle filtering. The input to this module include the amplitude, phase, time delay, planar incident angles, and Doppler frequency shift of each of the multipath components. The net output from the seeker module is two planar angles in azimuth and elevation which describe the orientation of the line-of-sight (LOS) of the vehicle to the estimated location of the target. These are used in the guidance computations.

The simulation driver calls HOJ which in turn calls the subroutines which model the signal processing. Subroutine HOJ in turn calls RDPHAS and DLS if specified in the EXEC. Subroutine RDPHAS adds a random phase between - and 2π to the phase of each of the multipath components. Subroutine DLS calls C1HOJ which models the enhanced interferometer processor. This DF estimator generates unbiased estimates on the strongest emitter in the field-of-view by performing an average of the phase difference between two pairs of antenna elements which are used to find elevation and azimuth conical angles. Note that the subroutine GAIN called by C1HOJ models the FHT antenna characteristics given the angle of arrival of the multipath signals. The details of the C1HOJ subroutine are discussed in Subsection 4.2 below. The outputs of the signal
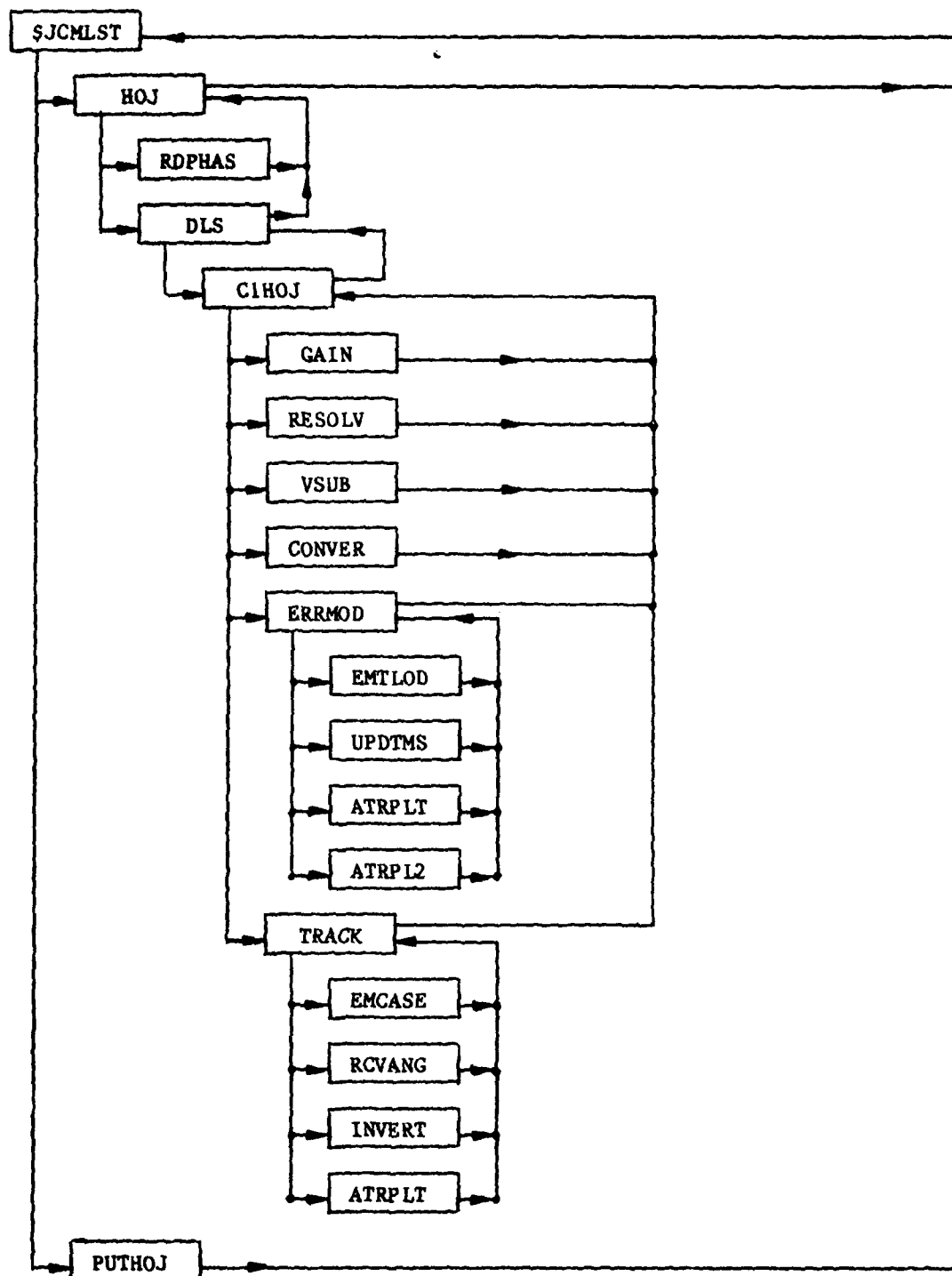
Fig. 4.1.   Seeker program sequence.

processing block are planar angles which are corrupted by errors which are characterized in ERRMOD. This subroutine models DF angle biases, antenna measurement errors and blind range effects. If specified, the corrupted planar angle estimates are filtered by the tracker algorithm in subroutine TRACK. Further description of the tracker algorithm is given in subsection 4.3 below. Finally, subroutine PUTHOJ is called to write out the vehicle position, velocity, boresight coordinates, planar angle estimates from the tracker, error angles, and planar difference angles to the disk.

4.2 Enhanced Interferometer Signal Processing

The signal processor is based on the phase interferometer concept. The module computes the real and imaginary parts of each signal present in the electromagnetic environment, at each of the antenna elements. The number of signals that are modelled is dependent on the number of emitters present and the number of multipaths associated with each emitter using amplitude and phase data. The phase is calculated using quantities characterizing multipath phases, time delays, fractional doppler frequencies, and the changes in phase due to differential path length. Changes in phase due to the characteristics of the receive antenna can also be incorporated. The real and imaginary parts of each signal sensed by the antenna elements are calculated using the phase and amplitude of each multipath component, and the gain characteristics of the receive antenna. Antenna characteristics are modelled by calling subroutine GAIN. This call is only implemented if IGAIN is 1 in BDUHFM. The gain is, of rse, dependent on the angle of arrival of each multipath. The processing scheme employs averaging the phase difference over the 50 samples taken in 0.1 sec. Therefore, signal characteristics are determined for each sample in the 100 ms period by incorporating the appropriate doppler effects, and leaving all other multipath quantities unchanged. It should be noted that by modifying the overall simulation interval (quantity DRATE in block data file BDUHFM), and the number of phases generated per call (quantity PP in C1HOJG2), the realism of the model is increased. This occurs because the effect of changes in vehicle position and orientation on multipath attributes over the interval that the 50 signal samples are collected is better simulated.

The effects of measurement noise on the received signal in space can also be modelled. The noise has a Gaussian distribution. The magnitude of the noise is dependent on the distance of the vehicle from the emitter specified as No. 1.

The phase difference (for each of the 50 samples considered) between the azimuth (1 and 3) and elevation (2 and 4) elements is next calculated. The phase difference for the azimuth channel is resolved about the quantity AM15 (specified in block data file BDUHFM) and π. The phase difference between elements 2 and 3 is next determined.

The phase differences between elements (2, 3) and (2, 4) are used in combination to resolve ambiguities in the estimate of the phase difference between elements (1, 3). Phase differences are averaged over for azimuth and elevation channels. Phase differences are then resolved about the averages and π to remove ambiguities and then averaged again. Transformation to conical angles in pitch and yaw is performed using the known wavelength of the emitters and the antenna element separation. (This assumes perfect calibration of the antenna.) The conical angles are then converted to planar angles which are used in the models for tracker and guidance. Angle estimates are modified in a call to an error modelling subroutine prior to an optional call to the tracker (pre-filter). A block diagram of the processing is given in Fig. 4.2 for further clarity.

4.3 Tracker Algorithm

The tracker module serves to filter the planar angle estimates generated by the signal processor prior to their use in guidance software. The software performs a number of different functions. It compares the D.F. estimates angles vs. truth angles for each of the emitters that exist in the field. Planar angles which are body referenced (since the EHT seeker is body fixed) are transformed using a detailed Euler transformation matrix (or simpler methods) to space referenced angles. A number (N) of the space referenced elevation and azimuth angles are accumulated. The collected data is used to calculate 'best' tracker estimates of the target position using a 1 coefficient least

4-4

ENTER

```
          ┌──────────────────┐
          │     PERFORM      │
          │    AMBIGUITY     │
          │   RESOLUTIONS    │
          └──────────────────┘
                   │
          ┌──────────────────┐
          │   AVERAGE OVER   │
          │    NUMBER OF     │
          │    SAMPLES       │
          └──────────────────┘
                   │
          ┌──────────────────┐
          │    RESOLVE       │
          │     ABOUT        │
          │    AVERAGES      │
          └──────────────────┘
                   │
          ┌──────────────────┐
          │   AVERAGE OVER   │
          │    RESOLVED      │
          │   QUANTITIES     │
          └──────────────────┘
                   │
          ┌──────────────────┐
          │  TRANSFORM FROM  │
          │    PHASE TO      │
          │  CONICAL ANGLES  │
          └──────────────────┘
                   │
          ┌──────────────────┐
          │    TRANSFORM     │
          │   TO PLANAR      │
          │     ANGLE        │
          └──────────────────┘
                   │
          ┌──────────────────┐
          │  INSERT ERRORS   │
          │   IN PLANAR      │
          │ ANGLE ESTIMATES  │
          └──────────────────┘
                   │
                   └────────────────► EXIT
```
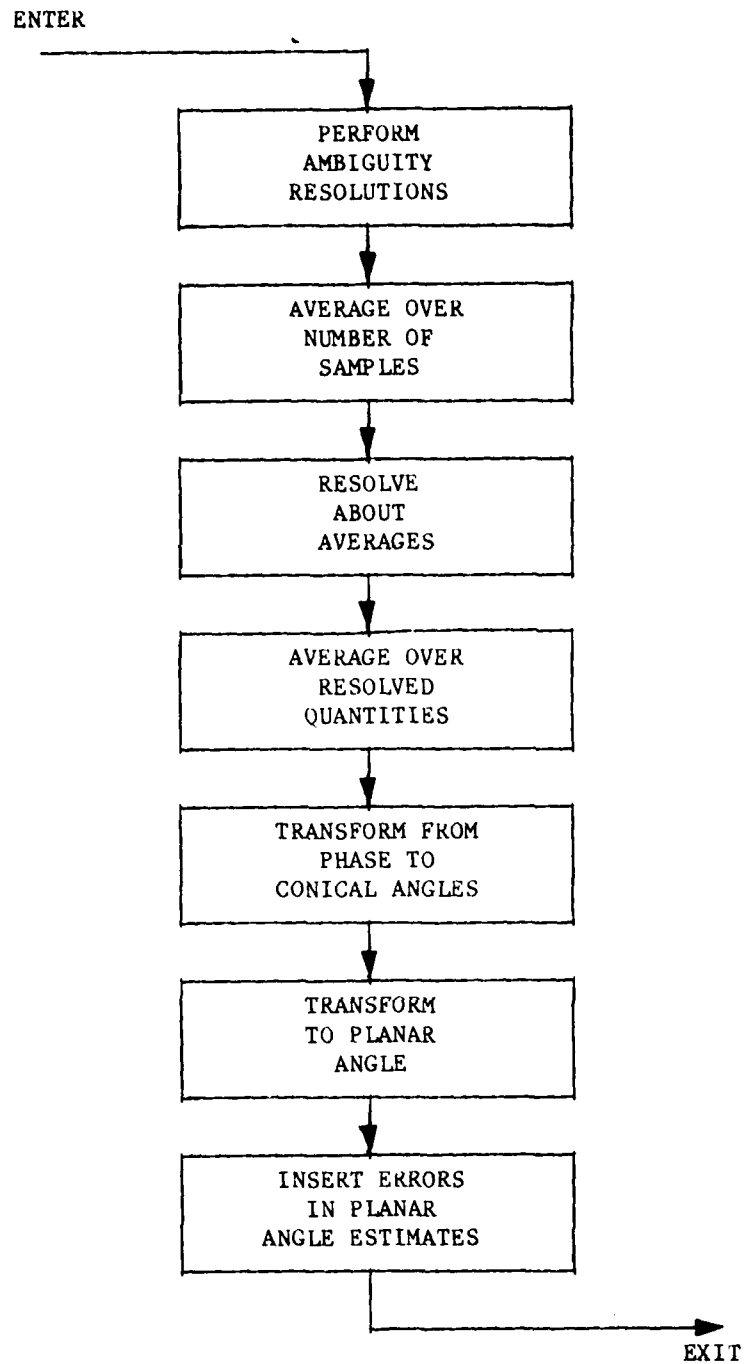
Fig. 4.2 Signal processing sequence.

4-5

squares fit. This fit is made either through a brute force calculation or through an iterative technique. Subsequent estimates (in both pitch and yaw) are declared good if the space referenced azimuth estimate passes a variance test which uses the magnitude of the variance and the 'best' tracker estimate in azimuth. If the estimate passes this test, it is used (together with the N-1 previous 'good' estimates) to rederive best tracker estimates and the variance magnitude. The original pitch and yaw body referenced estimates are passed to guidance without modification. If the azimuth angle fails the variance test, it (together with the corresponding elevation estimate) is replaced by angles computed by using the best tracker estimates and transforming back into the body frame. The tracker then enters coast mode. Coast logic depends on the state of the vehicle (altitude hold or pitchover initiated).

A.   Altitude Hold

For a finite period, ΔT, all estimates will be rejected and replaced by tracker generated angles. Following this interval, if N successive estimates pass a footprint test, and the variance of this data is small relative to a prespecified constant σ, the vehicle will reenter track mode and estimates will be passed unmodified to the guidance computer. (The footprint test determines whether the vehicle can home on the target corresponding to the new estimates.)

B.   Pitchover Initiated

If the vehicle is in the pitchover mode, the rejection of one set (Az, El) of estimates, does not automatically entail the rejection of subsequent estimates. 'Best' tracker estimates and the variance are frozen until N successive angles pass the variance criterion.

5.0 FLIGHT DYNAMICS MODULE

The program sequence for the flight dynamics module is shown in Fig. 5.1. As noted in Section 1.0 the call to the flight dynamics module, FLTDYN, is made from the master simulation driver $JCMLST. FLTDYN controls the sequencing of the other subroutines in the module:

CRSGID	Guidance subroutine
AUTPLT	Autopilot subroutine
AERDYN	Aerodynamics subroutine
EQAMD	Equations of motion subroutine
KINEM	Kinematics subroutine

As may be seen in the diagram some of these subroutines call still other subroutines. This section will discuss in a general way the purpose of these subroutines, while Appendix D will provide additional details of the input and output, as well as the definition of the variables as given in the program listings which accompany this report.

One of the first tasks of the FLTDYN subroutine is to initialize the parameters in the subroutines which it calls. In particular, it calls separately the subroutine WIND, which models the wind shear and turbulence to which the vehicle is exposed. FLTDYN also controls the cycling interval of the various subroutines. For example, the cycling rate for the guidance computer (CRSGID) is slower than for the other subroutines called. In addition, it is to be noted that the cycling rate for the FLTDYN subroutine will, in general, be different than for the subroutines of the multipath part of the simulation discussed in Section 3.0.

After initialization, the first subroutine to be called is CRSGID, the guidance computer module. The key inputs to the guidance computer are the planar angle measurements which specify the orientation of the vehicle with respect to the estimated line-of-sight (LOS) to the target and the pitch and yaw body angular rates. These rates are filtered and combined to determine
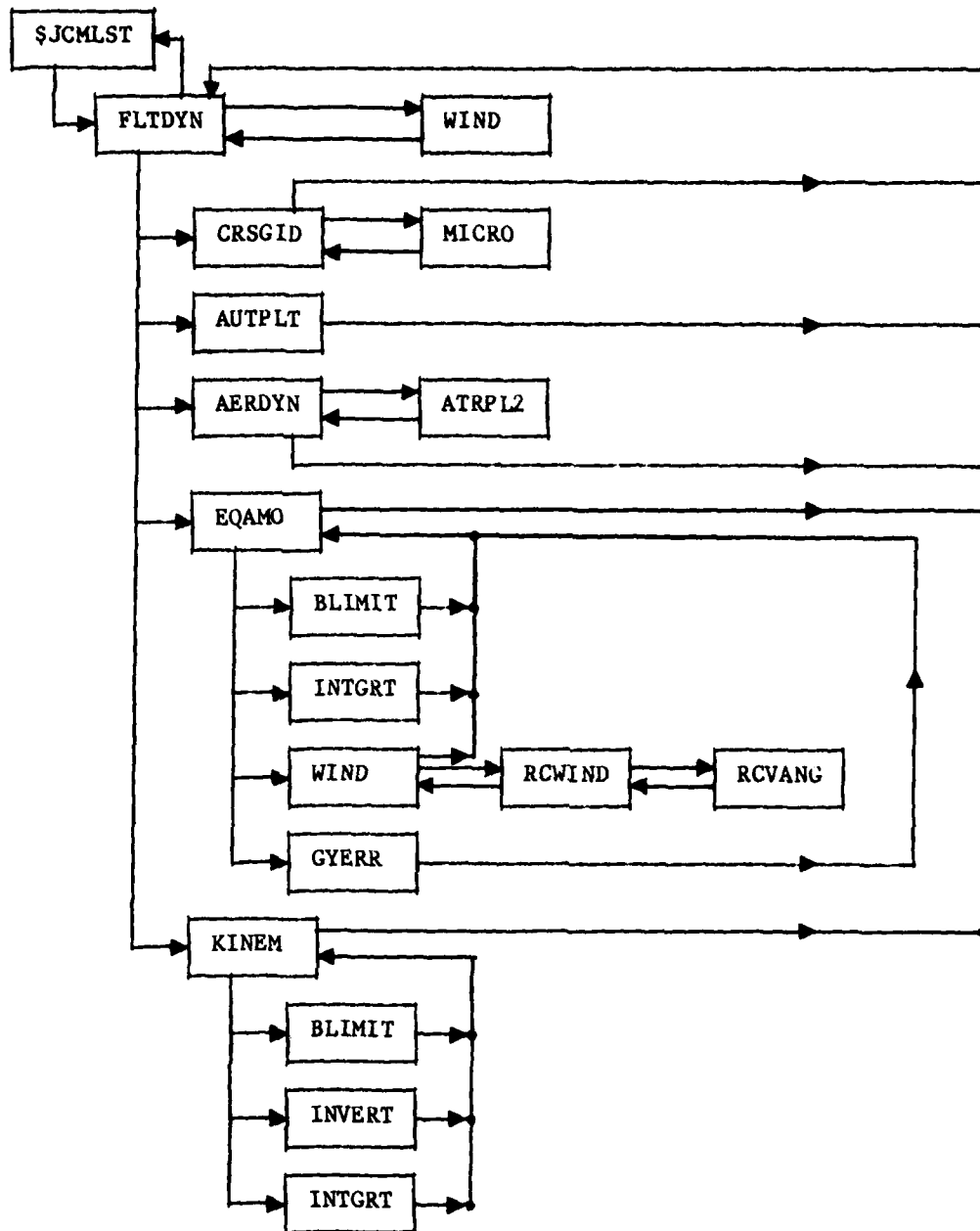
5-1

Fig. 5.1. Flight dynamics program processing.

pitch and yaw commands to the autopilot using both pursuit and proportional navigation laws. Filtered planar angles are also used as a basis for specifying mode switch conditions to the autopilot to transition from altitude hold to dive on the simulated target. Several seconds after pitchover initiation, a guidance mode switch occurs and the vehicle transitions from pursuit to proportional navigation. The guidance computer also calls the subroutine MICRO which simulates the effects of the actual word length used in the microcomputer on which the actual guidance computer is implemented.

The next subroutine called by FLTDYN is the autopilot module. The key inputs to the autopilot module are the pitch and yaw navigation commands, the mode switch commands and the pitch and yaw body rates. The commands are provided by the guidance module while the pitch and yaw body angular rates are computed in the equation-of-motion module (EQAMO). The outputs are the control surface deflections which specify the orientation of the elevator, rudder and ailerons. It is to be noted that the pitch and yaw body rates provided to both guidance computer and autopilot are corrupted with errors which model the actual gyro performance.

After return from the autopilot module, FLTDYN calls the aerodynamics module, AERDYN, which computes the aerodynamic coefficients and thrust on the vehicle. The aerodynamic coefficients are nonlinear functions of many variables including angles-of-attack and sideslip, control surface deflection, body rates, and thrust. In addition, the linear and angular velocities associated with the wind field, which are computed in the equations-of-motion module, are also input. It is further noted that the module also linearly increases the RPM of the engine after pitch over to reflect the observed engine speed up that occurs in a dive. Since the characteristics of the thrust are given in a two-dimenstional table (RPM, velocity) a two-dimensional table look-up routine, ATRPL2, is called by AERDYN.

After returning from the aerodynamics module FLTDYN calls the equation-of-motion module, EQAMO. This subroutine uses the aerodymamic coefficients produced by AERDYN and the inertial parameters of the vehicle to calculate the linear and angular accelerations. The angular accelerations are integrated to

provide angular rates (using INTGRT) which are then used to compute the Euler or space angular rates. The Euler rates are then integrated (again using INTGRT) to give the Euler angles of the body defining its orientation in space. Also in this module the linear and angular components of the wind field in the vehicle coordinate system are computed from the subroutine WIND. In accomplishing this task, WIND calls RCWIND which transforms the inertially referenced wind field into body reference components. In turn RCWIND calls RCVANG which uses inertially referenced coordinates to determine the azimuth and elevation angles with respect to the vehicle.

The BLIMIT call used by the EQAMO subroutine is used to limit the elevation Euler angle (θ) from coming too close to 90 degrees ($\pi/2$) since in the computation of Euler angle a division by $\cos\theta$ is made.

Finally, a subroutine to compute gyro errors GYERR is called. This subroutine provides the measurement of pitch and yaw body rates corrupted by bias, scale factor, and minimum detectable signal level which is used by both the guidance computer and autopilot.

The last call made by FLTDYN is to the subroutine KINEM which takes the linear velocities in the vehicle frame and transforms them to linear velocities in the inertial frame using a transformation matrix computed from the Euler angles (a call to INVERT). These vehicle space velocities are integrated (a call to INTGRT) to give vehicle space positions.

6.0 THE USE OF THE EMITTER HOMING SIMULATION

The emitter homing simulation is run in time-share mode by typing in the EXEC,

TESTSMM EXEC A

This EXEC, together with the EXEC TESTSIM2, loads the key programs for the execution of the simulation. The EXEC TESTSIM2 also serves to specify the initial conditions for the simulation case considered.

Most of the simulation programs do not model the tactical scenario, seeker processing, and vehicle flight dynamics. Those programs that characterize these parameters and critically affect the output of the simulation are listed below:

(1) BDUHFM

(2) GDDRONE

(3) GDAUT

(4) GDGUID

(5) CSRPVTRK

(6) ClHOJG2

(7) CCGAIND

Those programs required to run the EHT simulation are shown in Table 6.1. The outputs from the simulation program for a typical run (no wind effects, no antenna measurement errors, no gyro errors) are included in the next section.

TABLE 6.1
TERMINAL ACTIVITY WITH DEFINITIONS

```
. testsim exec a
EXEC TESTSIM2 BDUHFM GDORONE GDAUT GDGUID CATI CSRPVTRK CIHOJG2 CCGAIND

R; T=0.01/ .*1 12:22:57

.t testsim2 exec a
&CONTROL ALL
EXEC S
EXEC BLIPOF
*EXEC TDISK D
EXEC F1OTP TAPI
EXEC F14D A
EXEC F15D A
EXEC F16D A
EXEC CSGAIN
LOAD SJCMLST &1 &2 &3 &4 &5 &6 &7 &8 CRSPLT1 CRSPLT2 HOJRDT CLSLOOP MLTPLOT JCHOJ JCSCOPES PNT3D (NOMAP
&BEGSTACK
```
(A)   1000 ENTER RUN ID NUMBER
(B)   XBQM-106 RUN 350MHZ 8KSP 3EMITTERS REAL ESTIMATE DATA
(C)   YES DO YOU WANT SPECULAR GROUND REFLECTION ?(YES:NO)
(D)   TDG TYPE IN TDG TO DO INTERFEROMETER PROCESSING
(E)   END TYPE IN END TO TERMINATE RCVR TYPE INPUT
(F)   15000 -500 2500 TYPE IN INITIAL POSITION, FT
(G)   150 -2 12 TYPE IN SPEED,FT/SEC,MISSAZ,MISSEL, DEGREES
(H)   TEXC
```
      &END
      START
      FI 10 CLEAR
      FI 14 CLEAR
      FI 15 CLEAR

R; T=0.01/0.03 13:23:04
```

FILE DEFINITIONS:

SJCMLST	=	EHT SIMULATION DRIVER.
BDUHFM	=	BLOCK DATA FILE.
GDORONE	=	FLIGHT DYNAMICS DRIVER, AERODYNAMICS, EQNS. OF MOTION.
GDAUT	=	AUTOPILOT MODULE.
GDGUID	=	GUIDANCE MODULE.
CATI	=	SUBROUTINE TO CALL ENHANCED INTERFEROMETER MODULE.
CSRPVTRK	=	TRACKER.
CIHOJG2	=	ENHANCED INTERFEROMETER MODULE.
CCGAIND	=	RECEIVE ANTENNA MODULE.
CRSPLT1	=	PROGRAM FOR PLOTTING OF FLIGHT DYNAMICS AND SEEKER PARAMETERS.
CRSPLT2	=	PROGRAM FOR WRITING ONTO AND READING OFF DISK.
HOJRDT	=	SUBROUTINE FOR DETERMINATION OF WHEN TO STOP THE SIMULATION.
CLSLOOP	=	PROGRAM FOR WRITING OUT INPUT INFORMATION ON TACTICAL SCENARIO TO TERMINAL.
MLTPLOT	=	PROGRAM FOR PLOTTING OUT MULTIPATH QUANTITIES.
JCHOJ	=	PROGRAM FOR MODELLING OF TRANSMIT ANTENNA PATTERN.
JCSCOPES PNT3D	=	PROGRAMS FOR SETTING UP PLOTTING PROGRAMS.

ENTRY DEFINITIONS:

(A)   RUN IDENTIFICATION NUMBER
(B)   TITLE OF RUN
(C)   SPECIFICATION OF WHETHER TO USE SPECULAR GROUND REFLECTION
(D)   SPECIFICATION OF TYPE OF SIGNAL PROCESSING TO BE IMPLEMENTED
(E)   SPECIFICATION ON WHETHER TO LIMIT SIMULATION RUN TO ONLY ONE SIGNAL PROCESSING OPTION
(F)   INITIAL VEHICLE POSITION
(G)   INITIAL VEHICLE SPEED, AZIMUTH AND ELEVATION OFFSET ANGLES WITH RESPECT TO LINE
      OF SIGHT TO EMITTER 1.
(H)   TEXC INDICATES TYPE OF TERMINAL BEING USED

7.0 FINAL OUTPUT

When the vehicle has either impacted or the simulation time limit has been reached, final miss distances are written to the terminal and diagnostic plots are generated. A sample set of annotated plots is given in this section. Since graphics programs are not interchangeable between the Lincoln and other systems, the plotting programs listed below are not discussed in detail:

```
Subroutine MPOUT (A,B)
   Program MPOUT
```

This subroutine outputs the minimum miss distance for a flight path for the closed loop simulation.

```
Subroutine ARPTMP
   Program CLSLOOP
```

This subroutine plots the HOJ tactical scenario and gives the amplitude rankings of the multipath components.

```
Subroutine FLTPUT (A,B)
   Program CRSPLT1
```

This subroutine writes out flight diagnostic parameters such as guidance commands, accelerations, body rates, and Euler angles.

```
Subroutine PLOT3D
   Program CRSPLT1
```

This subroutine plots missile trajectories in three dimensions and in two dimensions.

```
Subroutine MLTPLT
   Program MLTPLOT
```

This subroutine plots out multipath diagnostics.

```
Subroutine DATPLT
   Program CRSPLT1
```

This subroutine plots out planar angles, planar projections of the vehicle's trajectory and velocity components.

```
PROGRAM TO DO CLOSED LOOP SIMULATION FOR EHT TACTICAL SCENARIO
1000    XBQM-106 RUN 350MHZ BKSP 3EMITTERS REAL ESTIMATE DATA
PARAMETERS FOR AZ    SYSTEM ARE:
PARAMETER          VALUE          UNITS         COMMENTS FOR CLARIFICATION

XMTRAZX            0.0            FT            X  )
XMTRAZY            0.0            FT            Y  } LOCATION OF EMITTER 1
XMTRAZZ            0.40000E+02    FT            Z  )
AZVELX             0.0            FT/S          X  )
AZVELY             0.42000E+04    FT/S          Y  } VELOCITY OF EMITTER 1 (IRREVELANT)
AZVELZ             0.0            FT/S          Z  )
WLAZ               0.28200E+01    FT            WAVELENGTH
PLZAZ                 T                         VERTICAL POLARIZATION
DIMAZ              0.0            FT            DIMENSIONS OF TRANSMIT ANTENNA
ITYPAZ                1                         EMITTER 1
PARAMETERS FOR DME    SYSTEM ARE:
PARAMETER          VALUE          UNITS

XMTRDMX            0.69280E+04    FT            X  )
XMTRDMY            0.40000E+04    FT            Y  } LOCATION OF EMITTER 2
XMTRDMZ            0.50000E+02    FT            Z  )
WLD               0.28100E+01    FT            WAVELENGTH
PLZD                  T                         VERTICAL POLARIZATION
DIMDE             0.40000E+01    FT            DIMENSIONS OF TRANSMIT ANTENNA (MUST BE NONZERO)
ITYPD                 2                         EMITTER 2
PARAMETERS FOR EL1    SYSTEM ARE:
PARAMETER          VALUE          UNITS

XMTRE1X            0.69280E+04    FT            X  )
XMTRE1Y           -0.40000E+04    FT            Y  } LOCATION OF EMITTER 3
XMTRE1Z            0.15000E+02    FT            Z  )
E1VELX             0.0            FT/S          X  )
E1VELY             0.0            FT/S          Y  } VELOCITY OF EMITTER 3 (IRRELEVANT)
E1VELZ             0.84000E+04    FT/S          Z  )
WLE1               0.28300E+01    FT            WAVELENGTH
PLZE1                 T                         VERTICAL POLARIZATION
DIMEL1            0.11820E+02    FT            DIMENSIONS OF TRANSMIT ANTENNA (IRRELEVANT)
ITYPE1                3                         EMITTER 3
PARAMETERS FOR EL2    SYSTEM ARE:
PARAMETER          VALUE          UNITS

XMTRE2X           -0.90000E+04    FT            X  )
XMTRE2Y            0.25000E+05    FT            Y  } LOCATION OF EMITTER 4
XMTRE2Z            0.15000E+02    FT            Z  )
E2VELX             0.0            FT/S          X  )
E2VELY             0.0            FT/S          Y  } VELOCITY OF EMITTER 4
E2VELZ             0.84000E+04    FT/S          Z  )
WLE2               0.26250E+01    FT            WAVELENGTH
PLZE2                 T                         VERTICAL POLARIZATION
DIMEL2            0.78720E+01    FT            DIMENSIONS OF TRANSMIT ANTENNA (IRRELEVANT)
ITYPE2                4                         EMITTER 4
```

Fig. 7.1. Transmitter parameters as listed by program with added comments.

PARAMETERS USED IN COMP OF SPEC GD REFL
ISPGRD- 1
ERO- 0.10E-01,-0.10E+00 SH2O- 1.700 NFZ- 2.80 NA-25 NB-11
DIFFUSE GD SCATTERING VALUES IN FT. ARE:

SYSTEM	SIGH	SIGL
AZ	0.5000	1.0000
DME	0.5000	1.0000
EL1	0.5000	1.0000
EL2	0.5000	1.0000

MULTIPATH EDITING PARAMETERS
THRESHOLDS FOR EACH PASS - 0.10E-04 0.10E-01 0.30E-01
OUT OF BEAMS: AZ: 3.00 DEG. EL: 3.00 DEG. DME: 0.50E-05 SEC
RUNLEN- 1000.00FT RUNWID- 100.00FT SAMPLING INTERVAL- 0.100 SEC

PT	X	Y	Z	VEL	BINC
1	21445.00	0.0	10084.00	1000.00	40.00
2	0.10	0.0	84.00	1000.00	200.00

XCUT IS -1.00

NOT USED IN CURRENT
ITERATION OF $JCMLST

ISPGRD = 1 :   ONLY A SIMPLE FRESNEL REFLECTION - ROUGHNESS COEFFICIENT IS TO COMPUTED

0 :   FULL INTEGRATION IS TO BE DONE

ERO   :   DEFAULT COMPLEX DIELECTRIC COEFFICIENT

SH2O   :   DEFAULT SQUARED RMS ROUGHNESS HEIGHT

NFZ   :   REAL NO. OF FRESNEL ZONES FOR GROUND REFLECTION INTEGRATION

NA   :   INTEGER NUMBER OF GRID POINTS ALONG MAJOR AXIS OF INTEGRATION ELLIPSE

NB   :   INTEGER NUMBER OF GRID POINTS ALONG MINOR AXIS OF INTEGRATION ELLIPSE

Fig. 7.2.   Multipath parameters with comments added for clarification.

7-4

BLINKING TRANSMITTER PARAMETERS ARE:

XMTR NO.	REL POWER LEVEL (DB)	REL VOLTAGE LEVEL	TRACK OPTION	AZ TRACK	EL TRACK	PATTERN	
1	0.0	1.00	FIXED PT	0.0	10.0	OMNI	(transmit antenna
2	0.0	1.00	FIXED PT	39.0	10.0	OMNI	pattern)*
3	0.0	1.00	FIXED PT	-23.0	10.0	OMNI	

3   CR TO GO ON..

RCVR TYPES ARE   .INTERFEROMETER

PARAMETERS FOR ENHANCED INTERFEROMETER RCVR MODEL ARE:

FAD2=   0.12800   CORRESPONDING TO 0.840 FT APERTURE IN AZ SYSTEM   (FAD2 = d/2 (AZ))*

FED2=   0.17920   CORRESPONDING TO 1.176 FT APERTURE IN EL SYSTEM   (FED2 = d/2 (EL))*

FIA =   1.88000   CORRESPONDING TO 0.840 FT APERTURE AT 350.3635 MHZ IN AZ SYSTEM   (FIA = $(2\pi d)/\lambda$ (AZ))*

FIE =   2.63000   CORRESPONDING TO 1.176 FT APERTURE AT 350.0974 MHZ IN EL SYSTEM   (FIE = $(2\pi d)/\lambda$ (EL))*

DYNX= 1.00   AM15= 0.0   ITRK= 0   IGAIN= 1   BWT= 1.325
(scaling parameter   (parameter for   (no tracker)*   (use CCGAIND)*
for elevation   resolution
planar angle)*   algorithm)*

INITIAL SPEED IS 150.0FT/SEC   MISSAZ IS -2.0DEG   MISSEL IS 12.0DEG
                                     (- implies velocity   (+ implies velocity
                                     vector is to the left   vector is above vector
                                     of vector to Emitter   to Emitter 1)*
                                     1)*

INITIAL POSITION (FT), VELOCITY VECTOR (FT/SEC) ARE:

PX = 15000.0   PY = -500.0   PZ = 1500.0   VX = -149.0   VY = -0.2   VZ = 16.9
(x vehicle   (y vehicle   (z vehicle   (x vehicle   (y vehicle   (z vehicle
position)*   position)*   position)*   velocity)*   velocity)*   velocity)*

NAZBLK = 0   NDMBLK = 0   NE1BLK = 0
                                                (no. of offtime
         1   2   3   intervals for
(emitter)*   (emitter)*   (emitter)*   emitter)*

┌─────────────────────────┐
│ *Note, comments are lower case │
│ and in ( ).   │
└─────────────────────────┘

Fig. 7.3. Receiver and vehicle parameters with comments added for clarification.
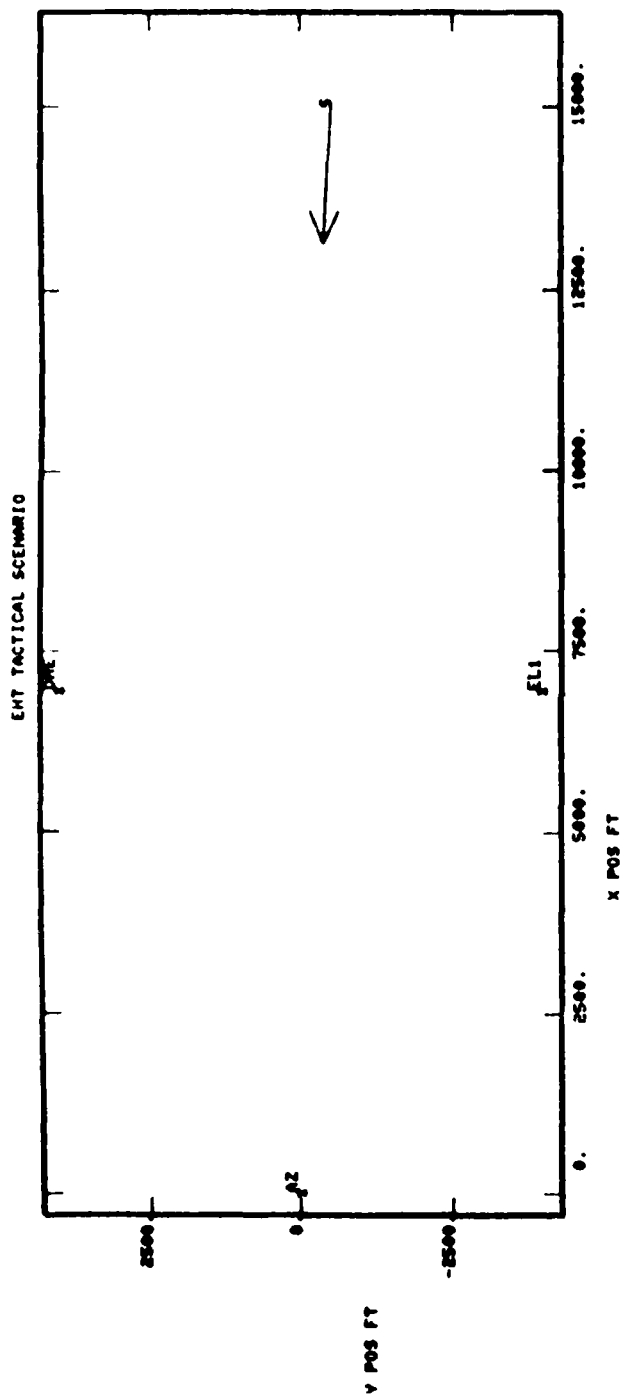
7-5

INPUTS FOR MODELING OF ERROR SOURCES

TYPE IN GYRO BIAS VALUE FOR AZ CHANNEL IN DEG/SEC
?
TYPE IN GYRO BIAS VALUE FOR EL CHANNEL IN DEG/SEC
?
TYPE IN 1 FOR SPECIFICATION OF MEASURED WIND DATA,0 OTHERWISE
?
TYPE IN WIND SPEED AT 30 FT ALT IN FT/SEC
?
TYPE IN WIND ANGLE W.R.T. TRUE EAST IN DEG
?
TYPE IN 1 FOR INCLUSION OF WIND GUST,0 OTHERWISE
?
TYPE IN RMS VALUE FOR MEAS. NOISE IN DEG
?
TYPE IN SEED FOR MEAS NOISE CALCS
?
TYPE IN OFF-BORESIGHT AZ ANGLE BIAS IN DEG
?
TYPE IN OFF-BORESIGHT EL ANGLE BIAS IN DEG
?
TYPE IN 1 IF YOU WISH TO ADD IN ANTENNA ERRORS,0 OTHERWISE
?
PROC
2

TIME	XPOS	VPOS	ZPOS	XVEL	YVEL	ZVEL
0.0	1500.0	-500.0	1500.0	-149.0	-0.2	16.9

TYPE IN SENSOR GAIN ERROR IN PITCH
?
TYPE IN SENSOR GAIN ERROR IN YAW
?
TYPE IN GYRO GAIN ERROR IN PITCH
?
TYPE IN GYRO GAIN ERROR IN YAW
?
TYPE IN ESTIMATED PITCH RATE CORRECTION IN DEG/SEC
?
TYPE IN ESTIMATED YAW RATE CORRECTION IN DEG/SEC
?

Fig. 7.4. Inputs for modeling of error sources.

CR TO BO SAT.

PROC	TIME	XPOS	YPOS	ZPOS	XVEL	YVEL	ZVEL

Fig. 7.5. Position and velocity coordinates, final miss distances, time used to perform multipath, receiver, vehicle computations.

USING
SUBROUTINE UPUTMS

USING
SUBROUTINE PTIME

7-7

EHT TACTICAL SCENARIO

Y POS FT

X POS FT

SIG PROC	EMITTER	OBST	RANK	AMP	TIME	RDOP
INT OP A	AZ	DRCT	4	0	1.4	0.0
INT OP A	DME	DRCT	1	-54	61.7	47.7
INT OP A	ELI	DRCT	3	-3	4.0	-5.1
INT OP A	AZ	GRND	5	4	61.8	0.3
INT OP A	DME	GRND	2	-28	61.8	-10.5
INT OP A	ELI	GRND	6	14	0.0	-5.8

SIG PROC : SIGNAL PROCESSING SCHEME EMPLOYED, INT OP A : ENHANCED INTERFEROMETER

EMITTER : WHICH EMITTER

OBST : DRCT = DIRECT WAVE
       GRND = GROUND REFLECTION

RANK : RANK OF MULTIPATH COMPONENT, ACCORDING TO AMPLITUDE WHICH IS WEIGHTED
       BASED ON RECEIVE ANTENNA PATTERN

AMP : AMPLITUDE IN DB OF MULTIPATH COMPONENT

TIME : TIME ASSOCIATED WITH MULTIPATH COMPONENT

RDOP : RECEIVER FRACTIONAL DIPPLER FREQUENCY THAT OCCURS AT MAX. WEIGHTED AMPLITUDE

Fig. 7.6. Schematic representation of EHT scenario.
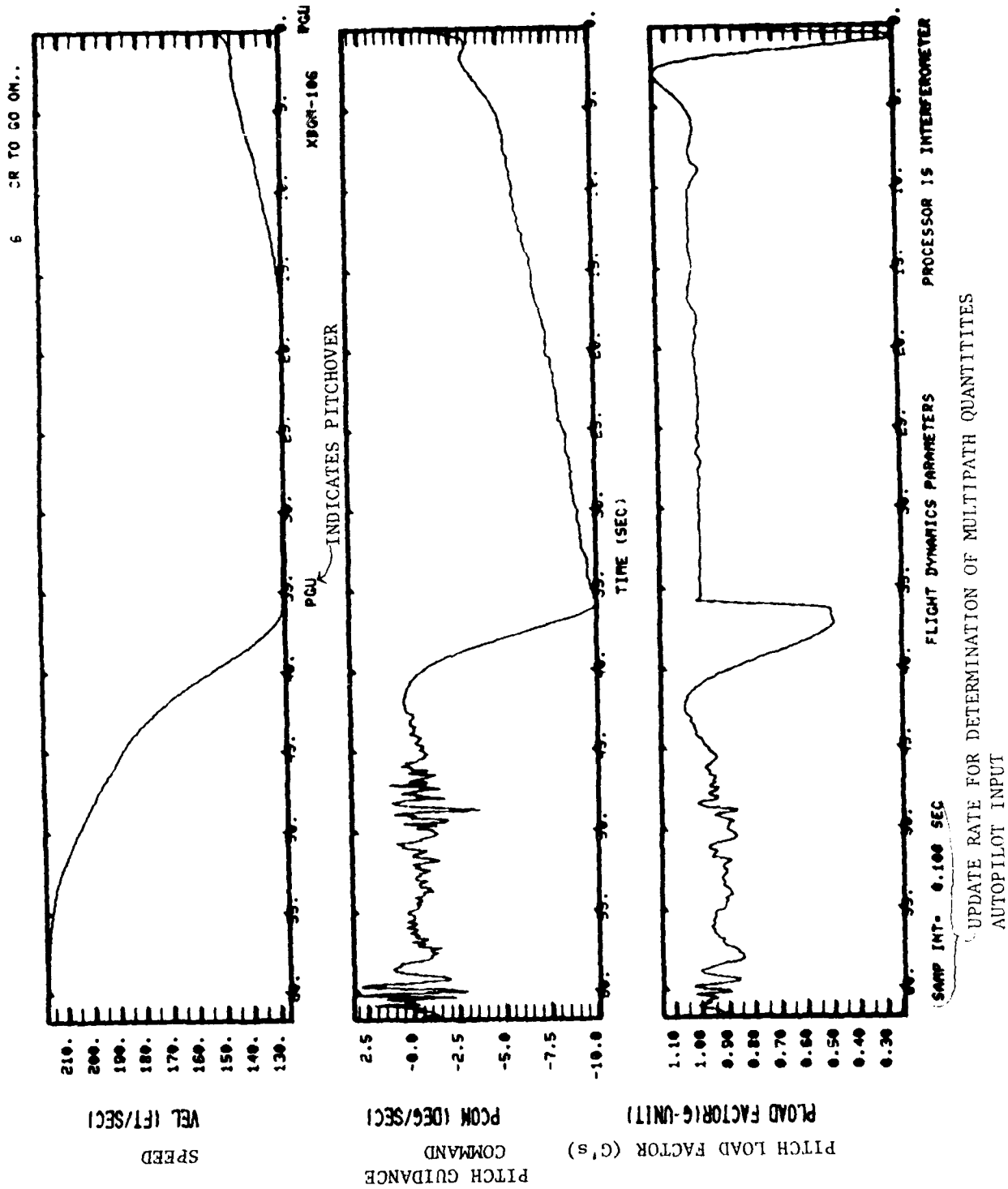
Fig. 7.7. Speed, Pitch guidance command, pitch load factor.
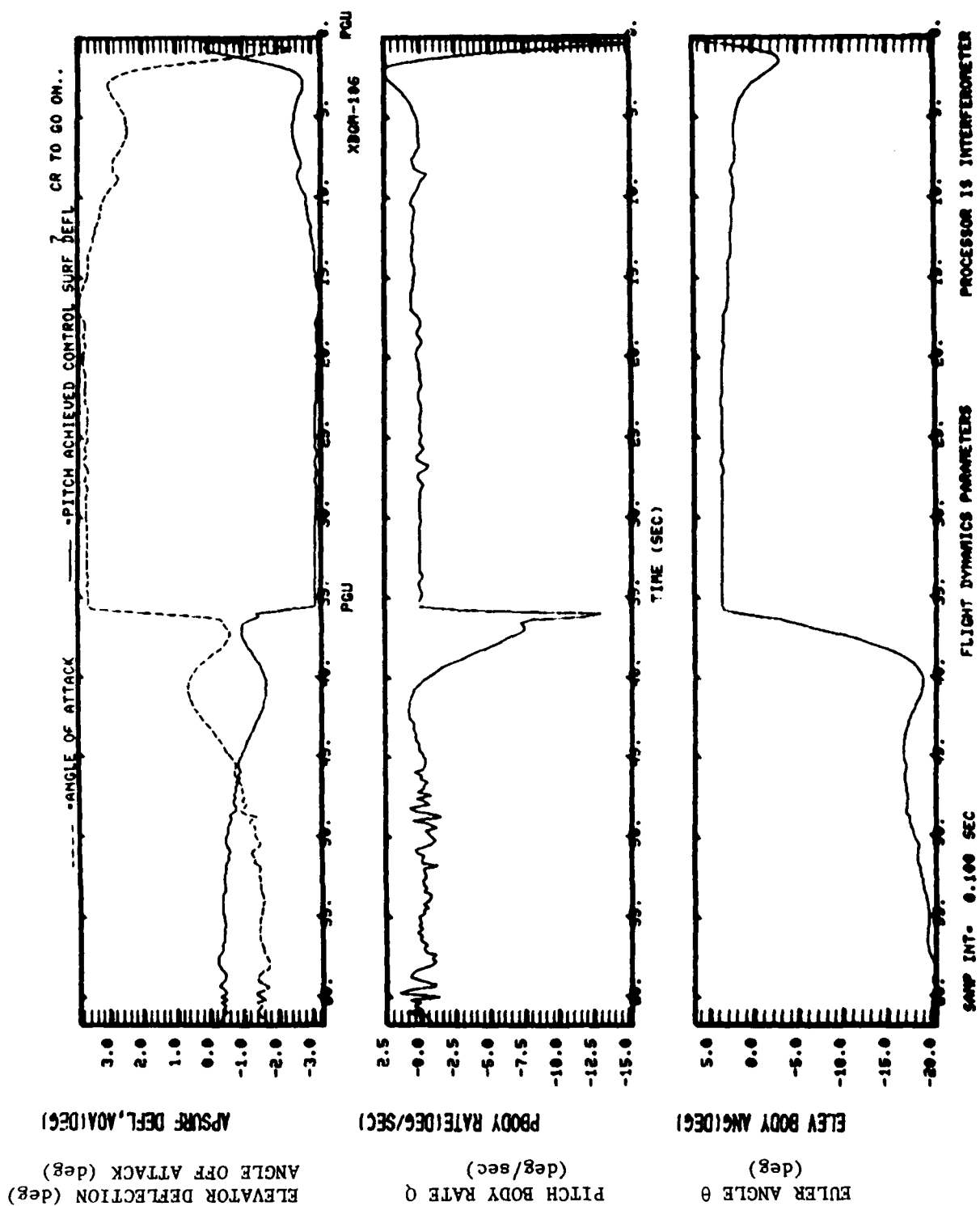
Fig. 7.8. Elevator angle/angle of attack, pitch body rate Q,
Euler angle θ.
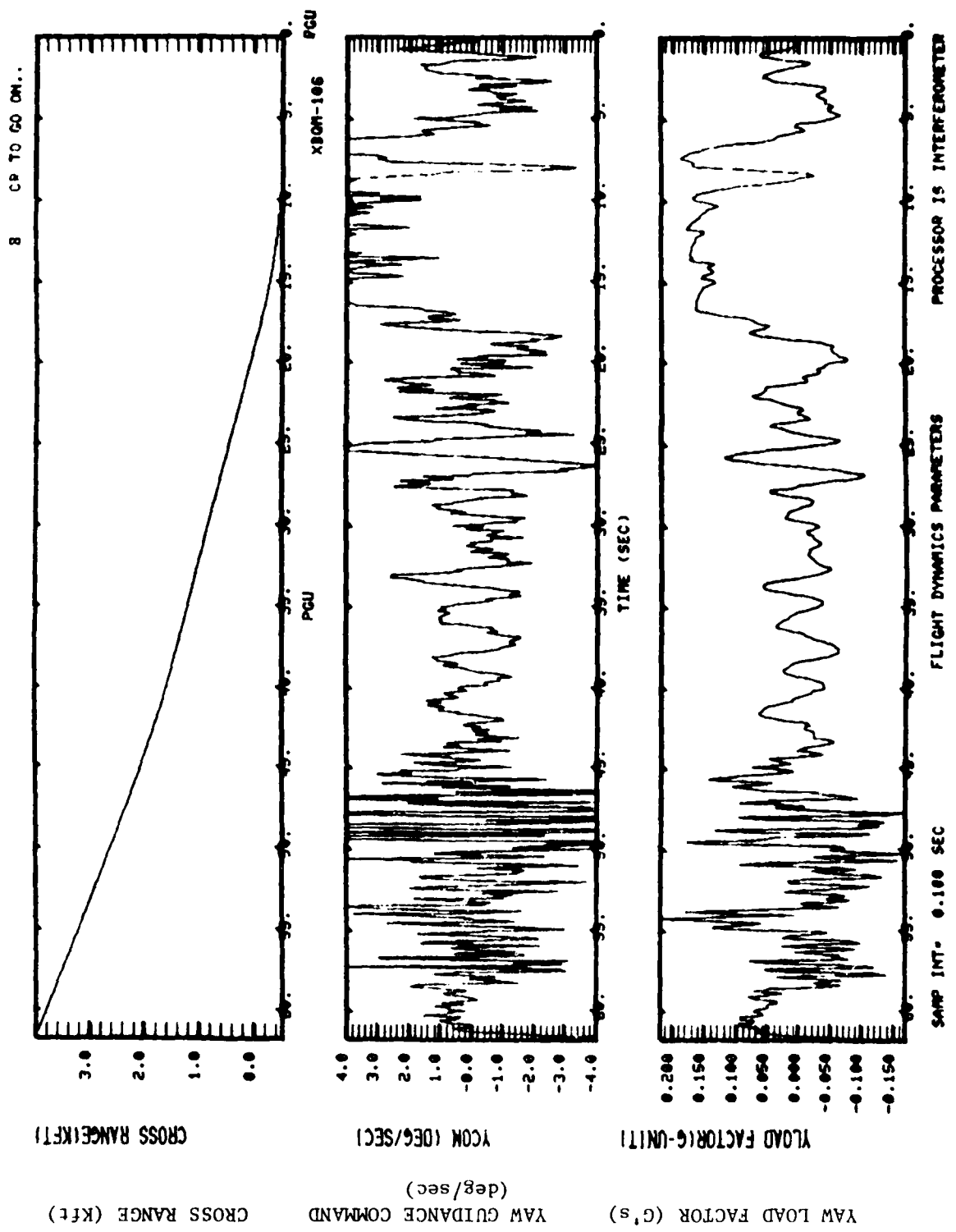
Fig. 7.9. Cross range, yaw guidance command, yaw load factor.
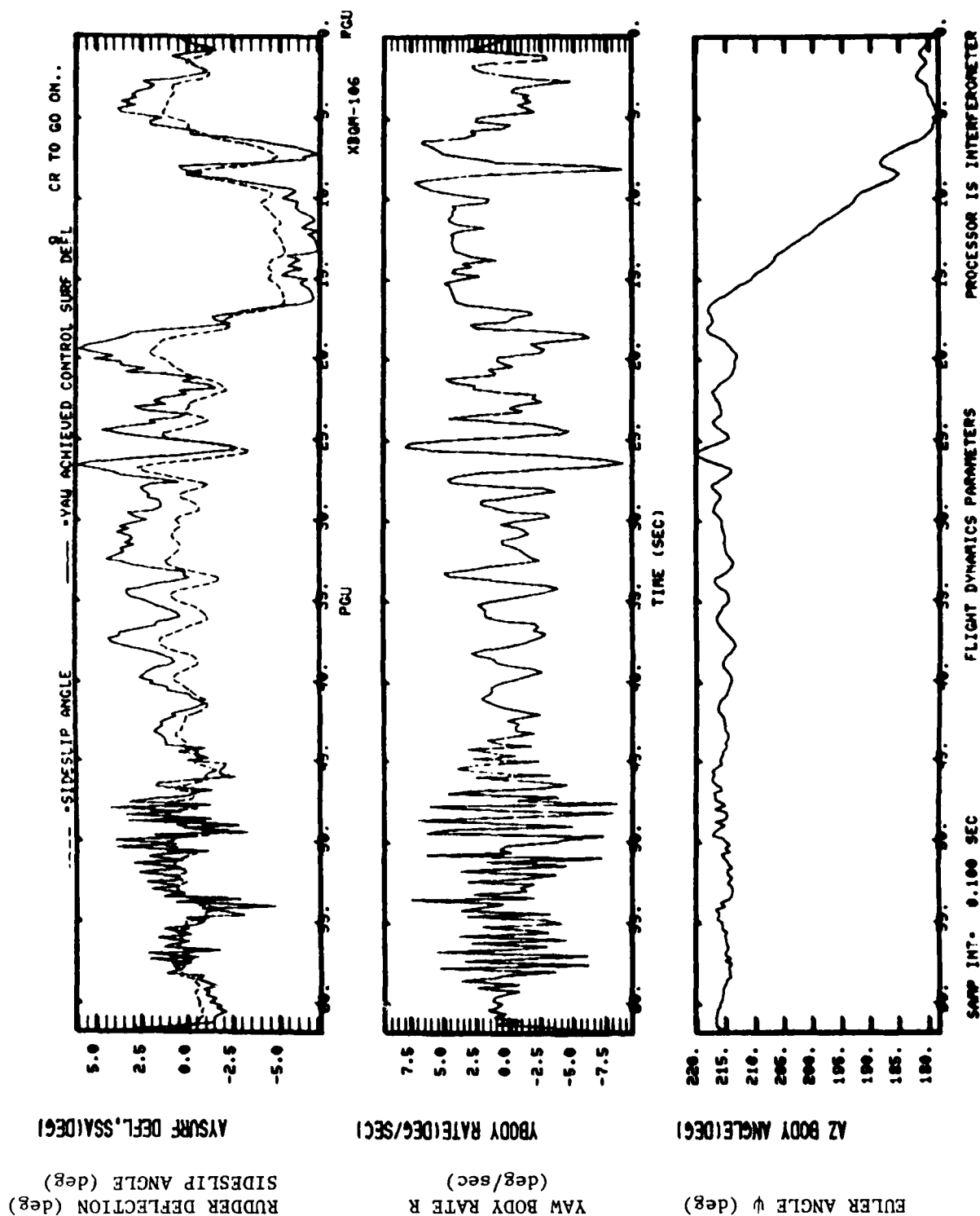
Fig. 7.10. Rudder deflection/sideslip angle, yaw body rate R,
Euler angle ψ.

7-12

Fig. 7.11. Additional elevation and azimuth errors, Elevation off-boresight angle input to guidance computer, azimuth off-boresight angle input to guidance computer.

7-13
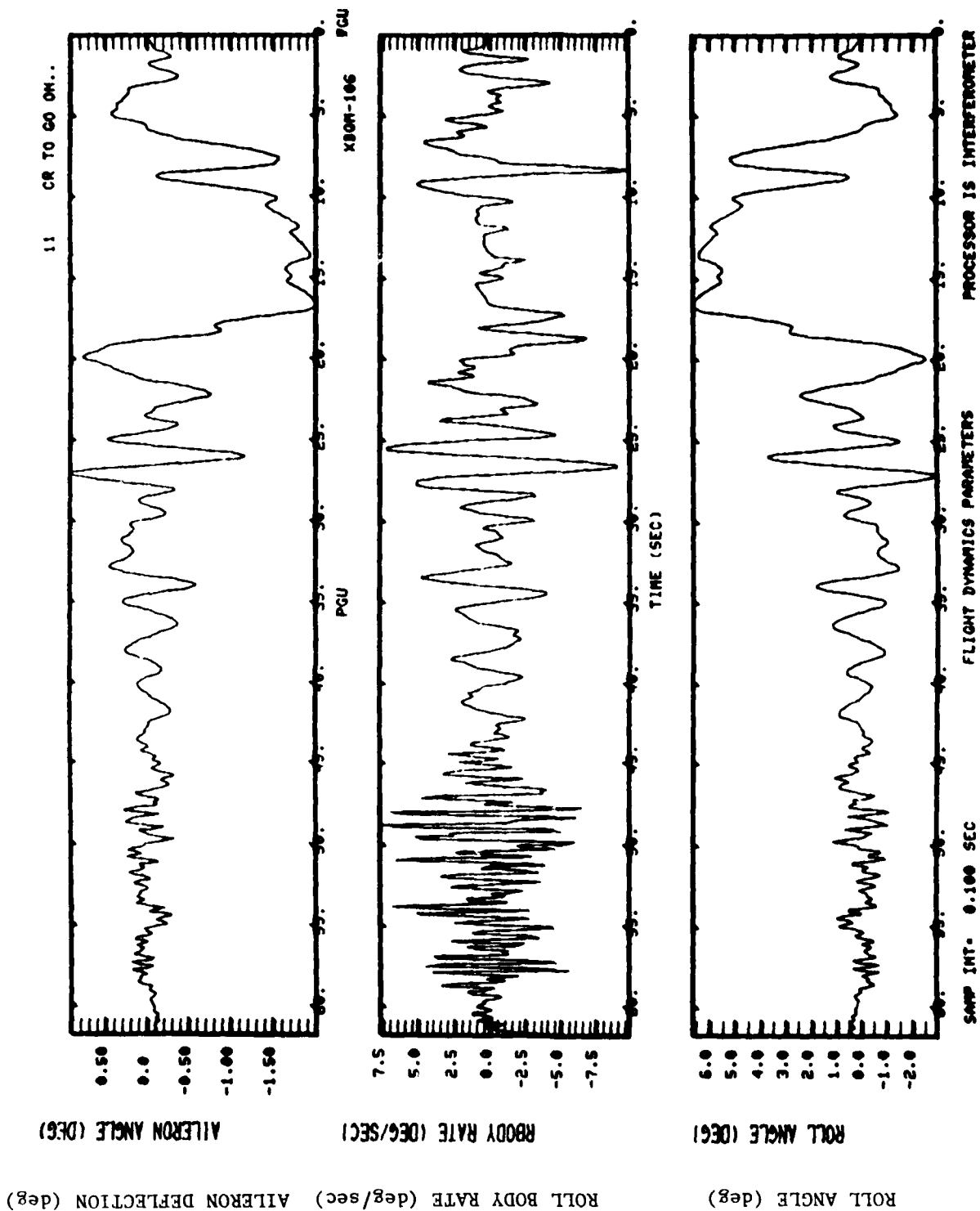
Fig. 7.12. Aileron angle (deg), roll body rate P (deg/sec), roll angle (deg).

7-14
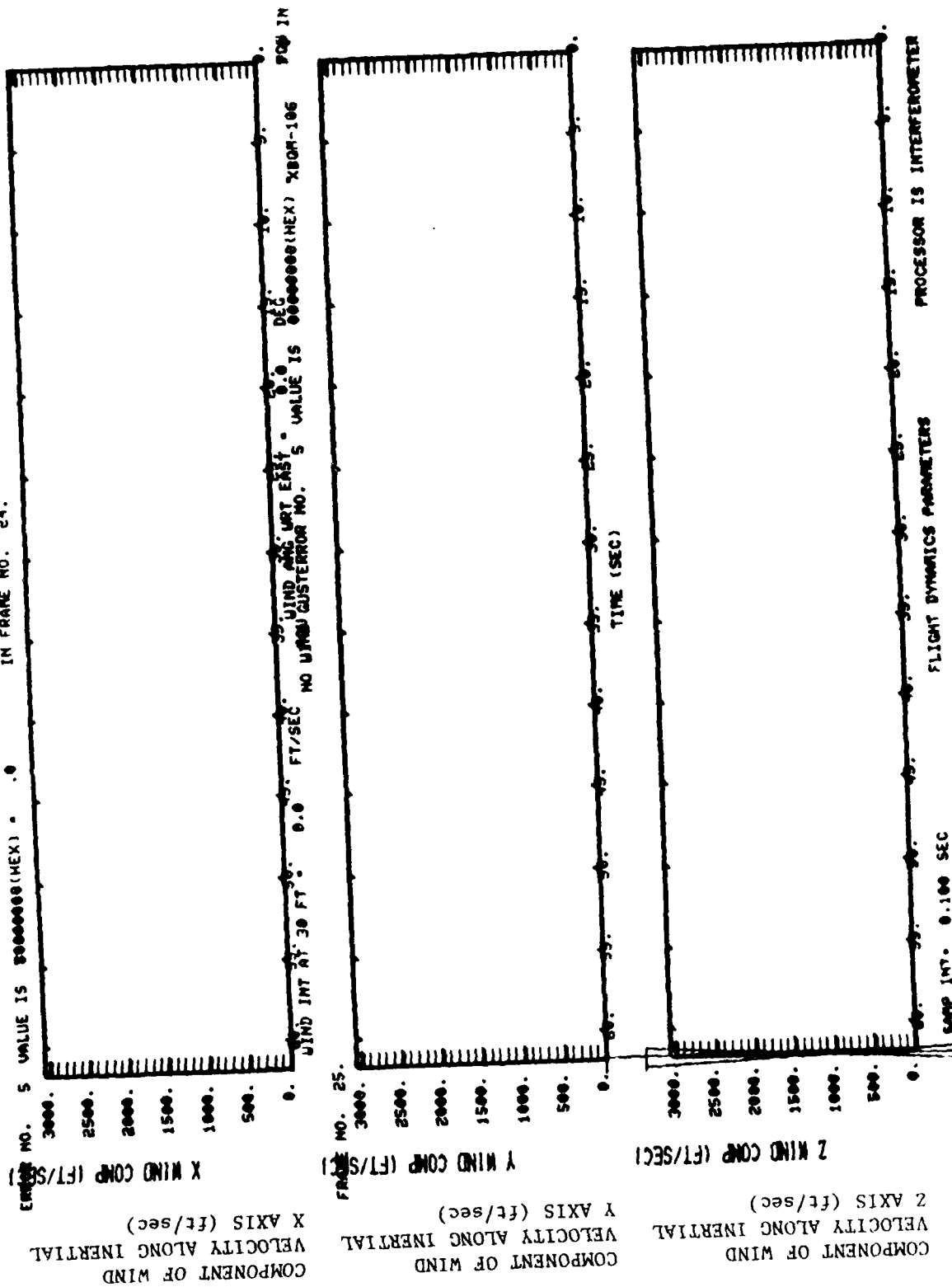
Fig. 7.13. Component of wind velocity along inertial X axis
component of wind velocity along inertial Y axis, and component
of wind velocity along inertial Z axis.
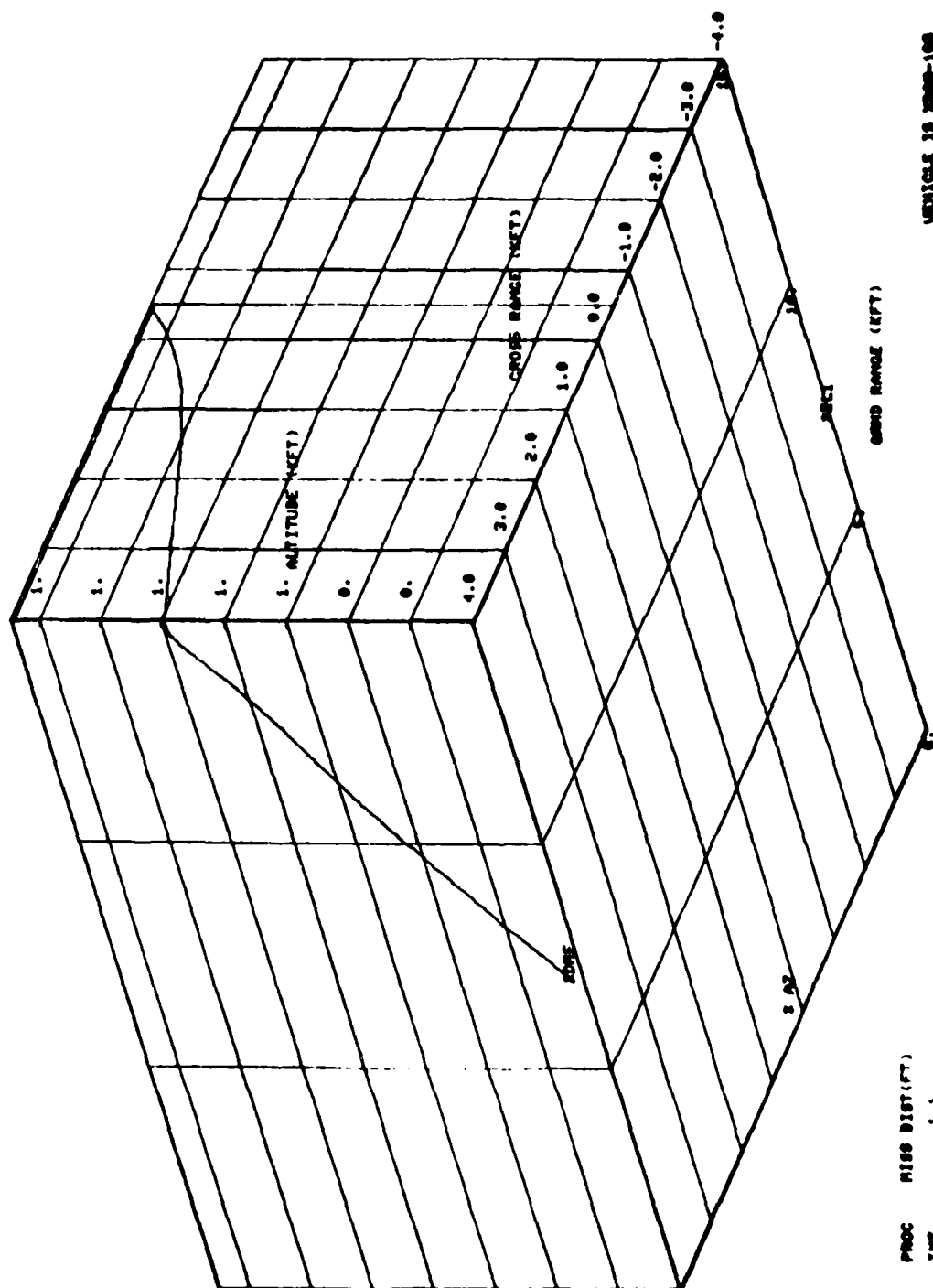
Fig. 7.14. Three-dimensional view of vehicle trajectory.
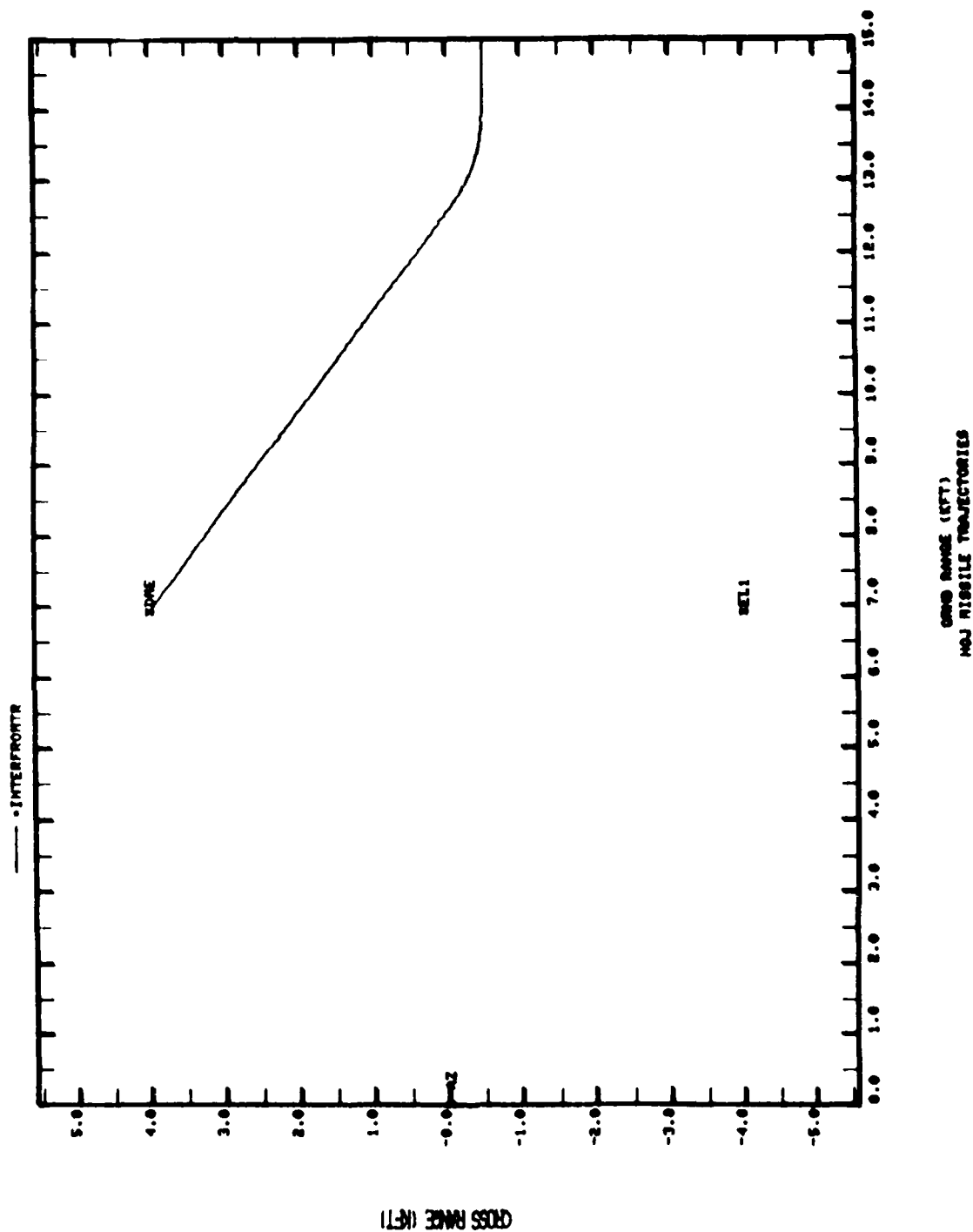
Fig. 7.15. Two-dimensional projection of vehicle trajectory.
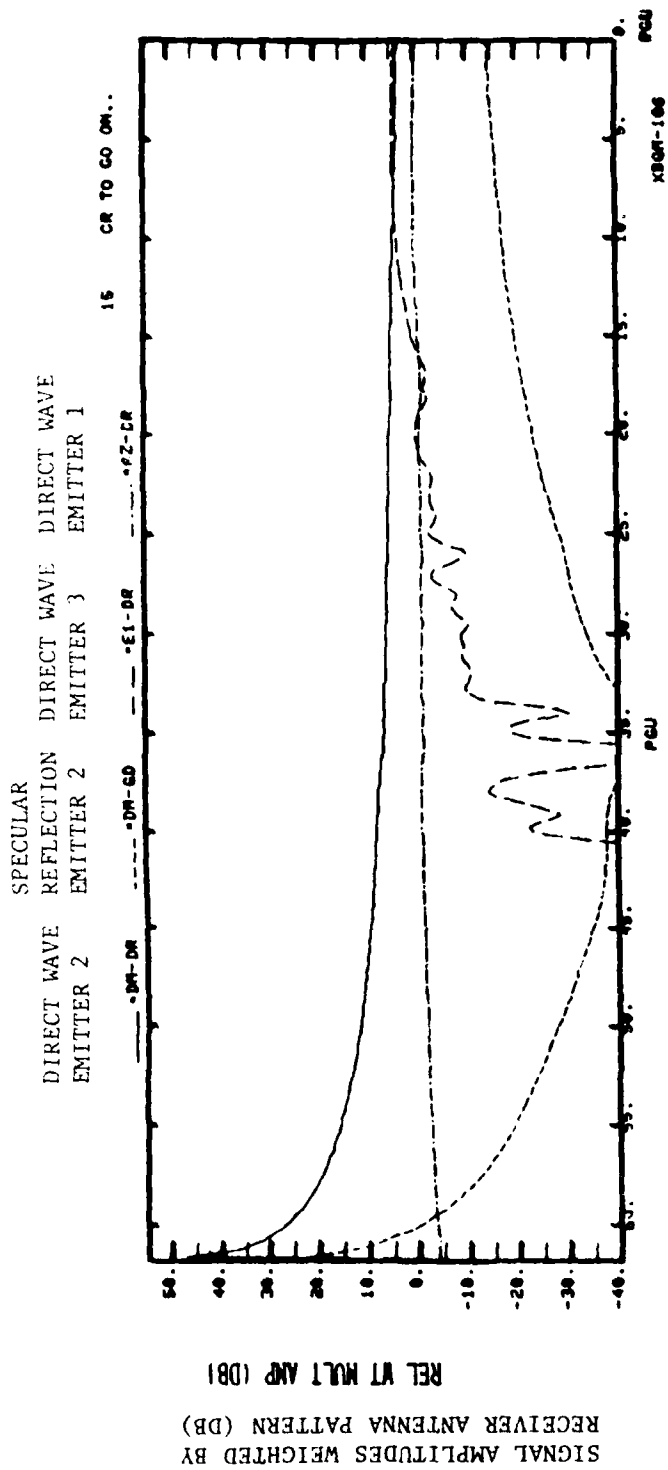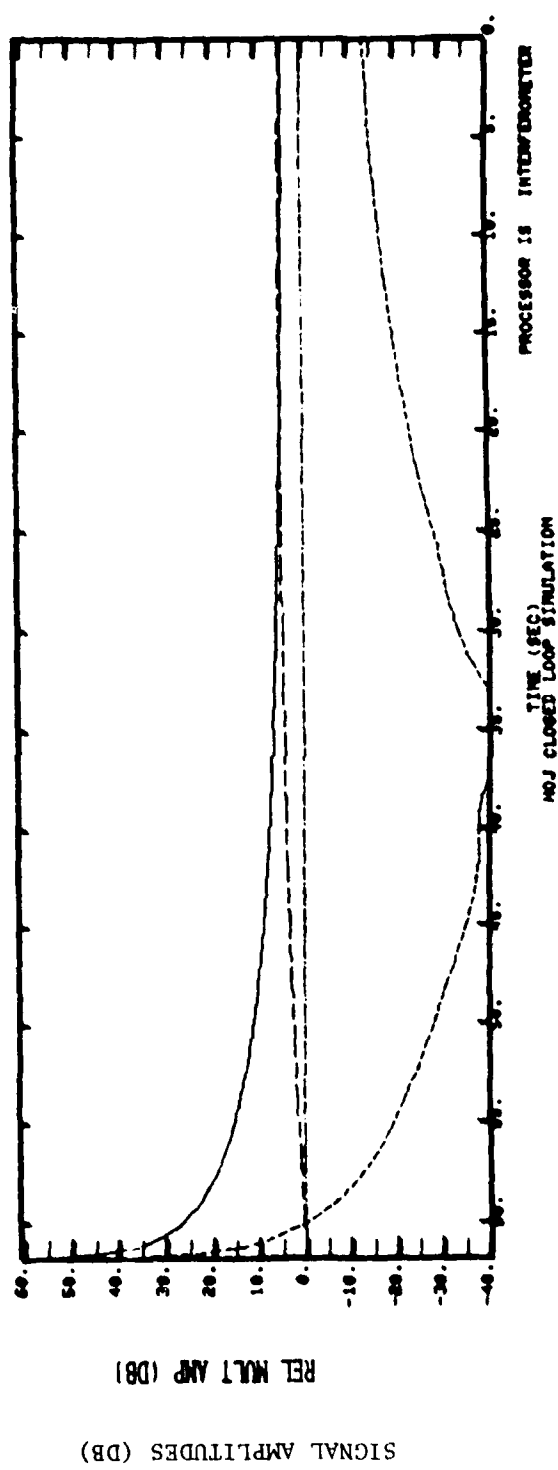
Fig. 7.16. Raw signal amplitudes, adjusted signal amplitudes incorporating receiver antenna pattern.
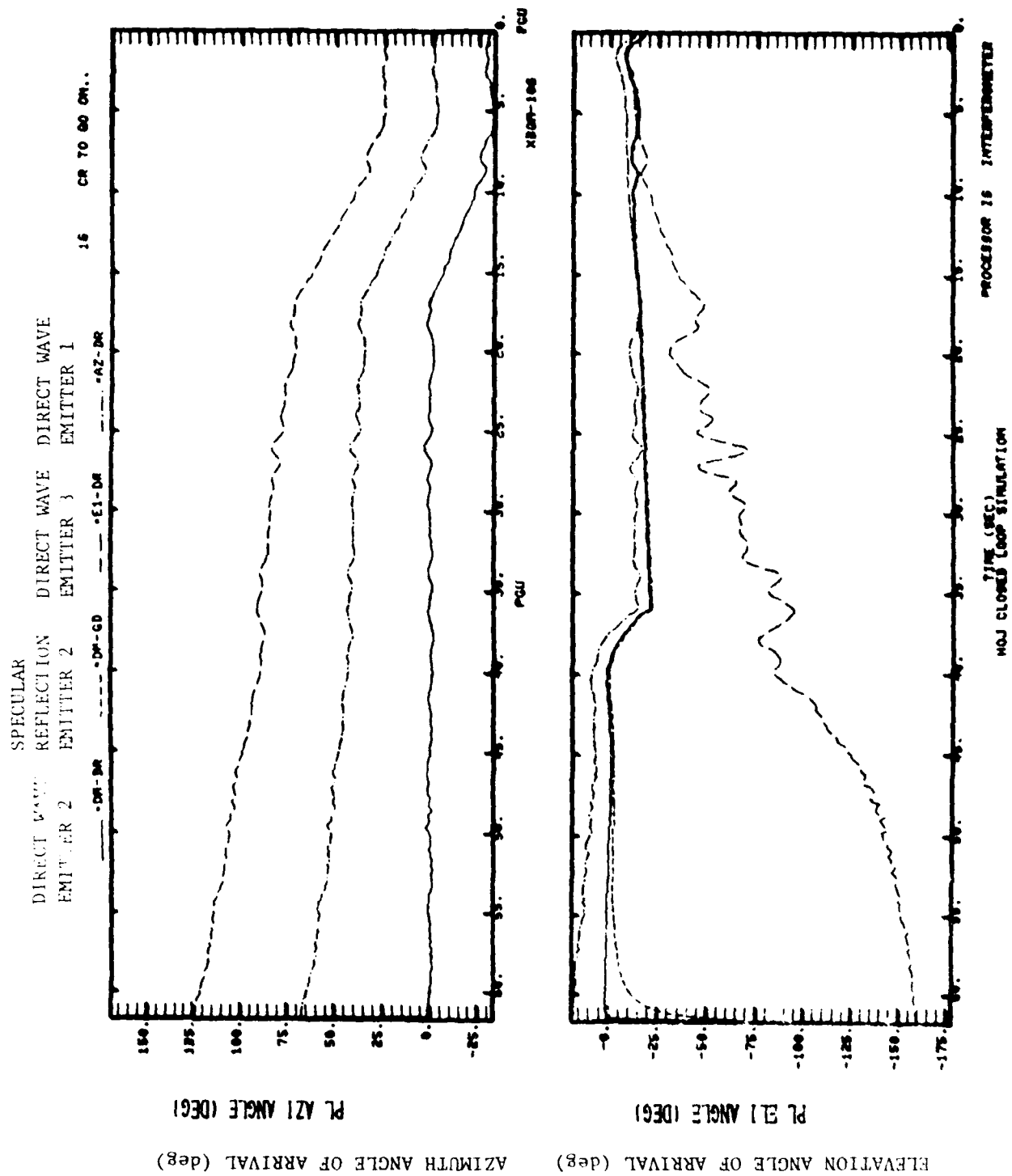
Fig. 7.17. Azimuth angle of arrival, elevation angle of arrival.

7-19
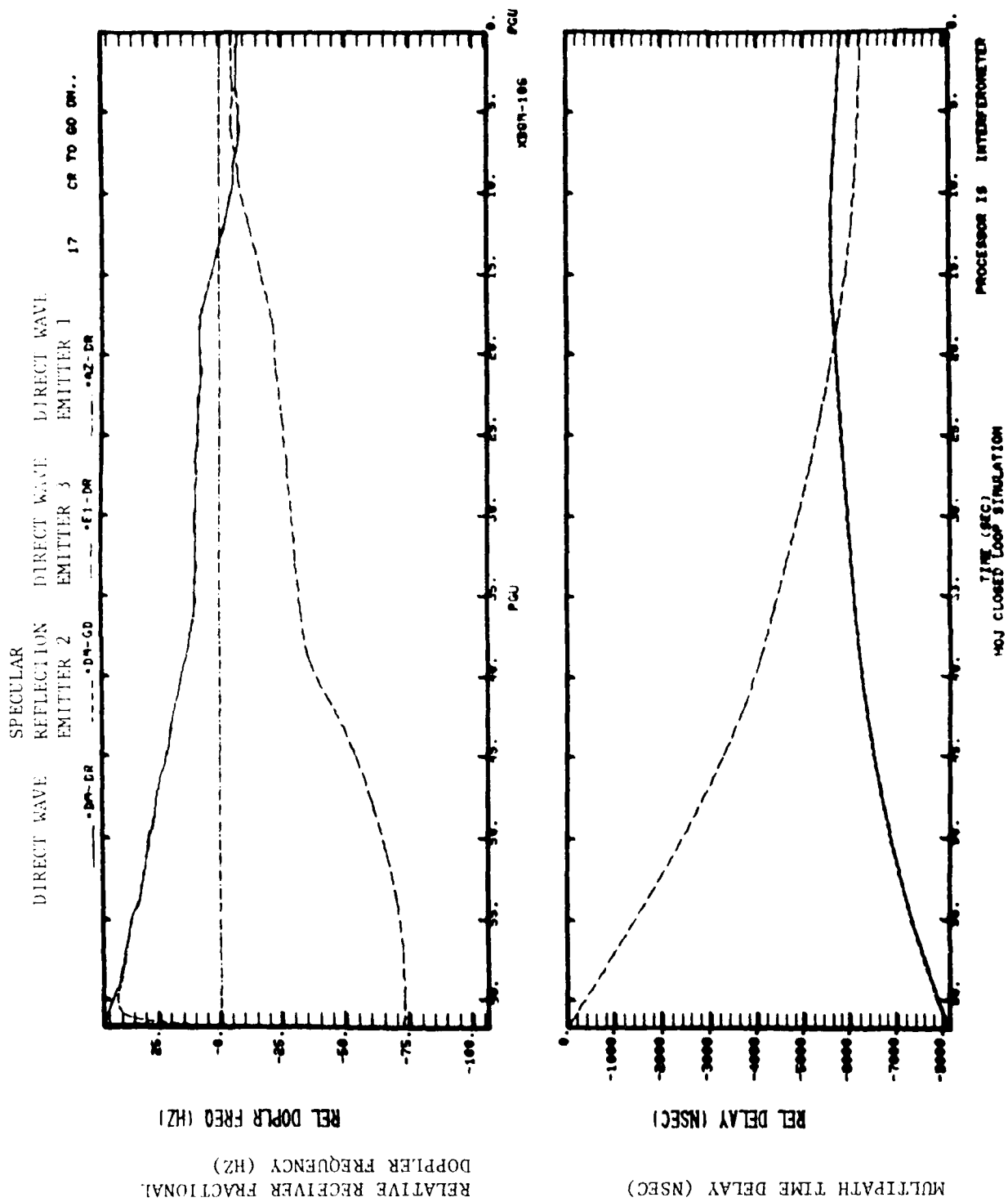
Fig. 7.18. Relative receiver fractional Doppler frequency, multipath time delay.
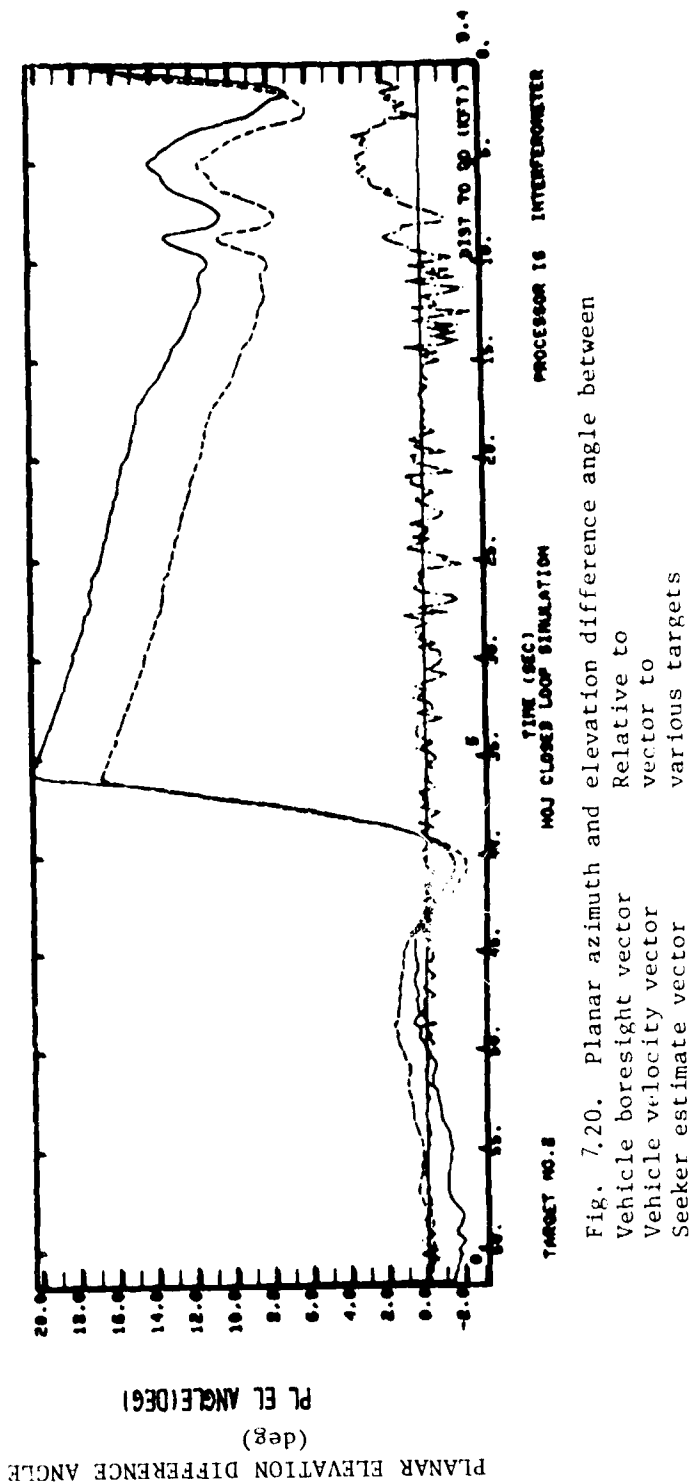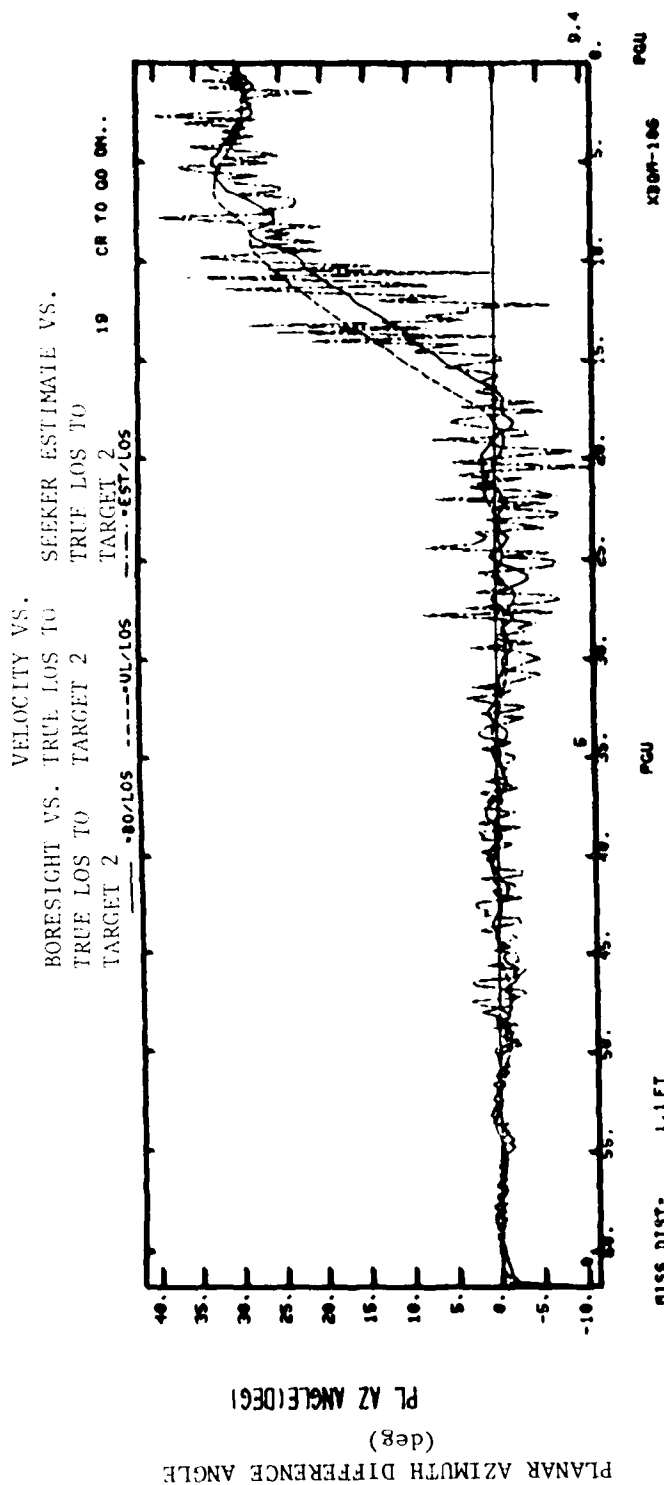
7-20

Fig. 7.19. Planar azimuth and elevation difference angle between
Vehicle boresight vector   Relative to
Vehicle velocity vector   vector to
Seeker estimate vector   various targets

Fig. 7.20. Planar azimuth and elevation difference angle between
Vehicle boresight vector   Relative to
Vehicle velocity vector   vector to
Seeker estimate vector   various targets

7-22
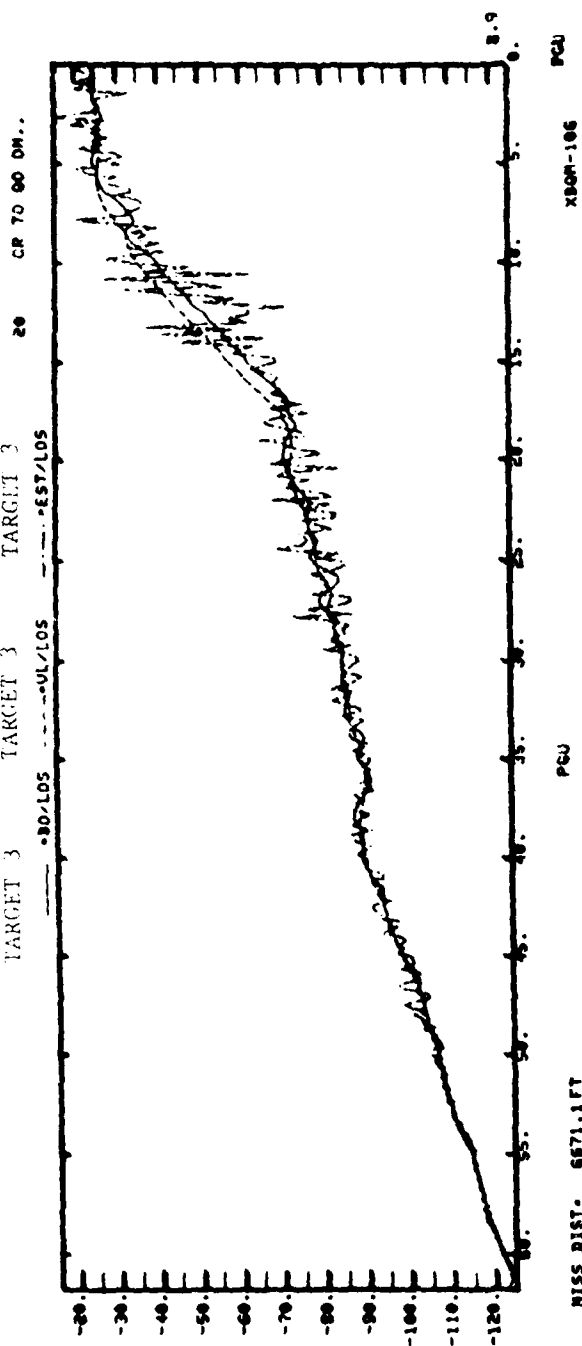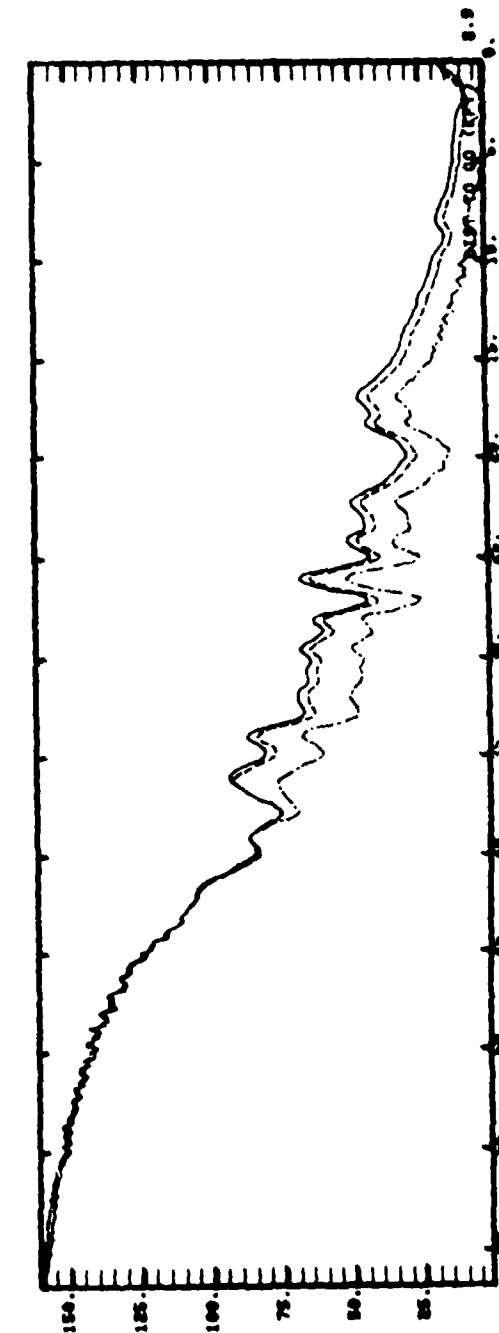
Fig. 7.21. Planar azimuth and elevation difference angle between
Vehicle boresight vector   Relative to vector
Vehicle velocity vector   to various
Seeker estimate vector   targets

7-23
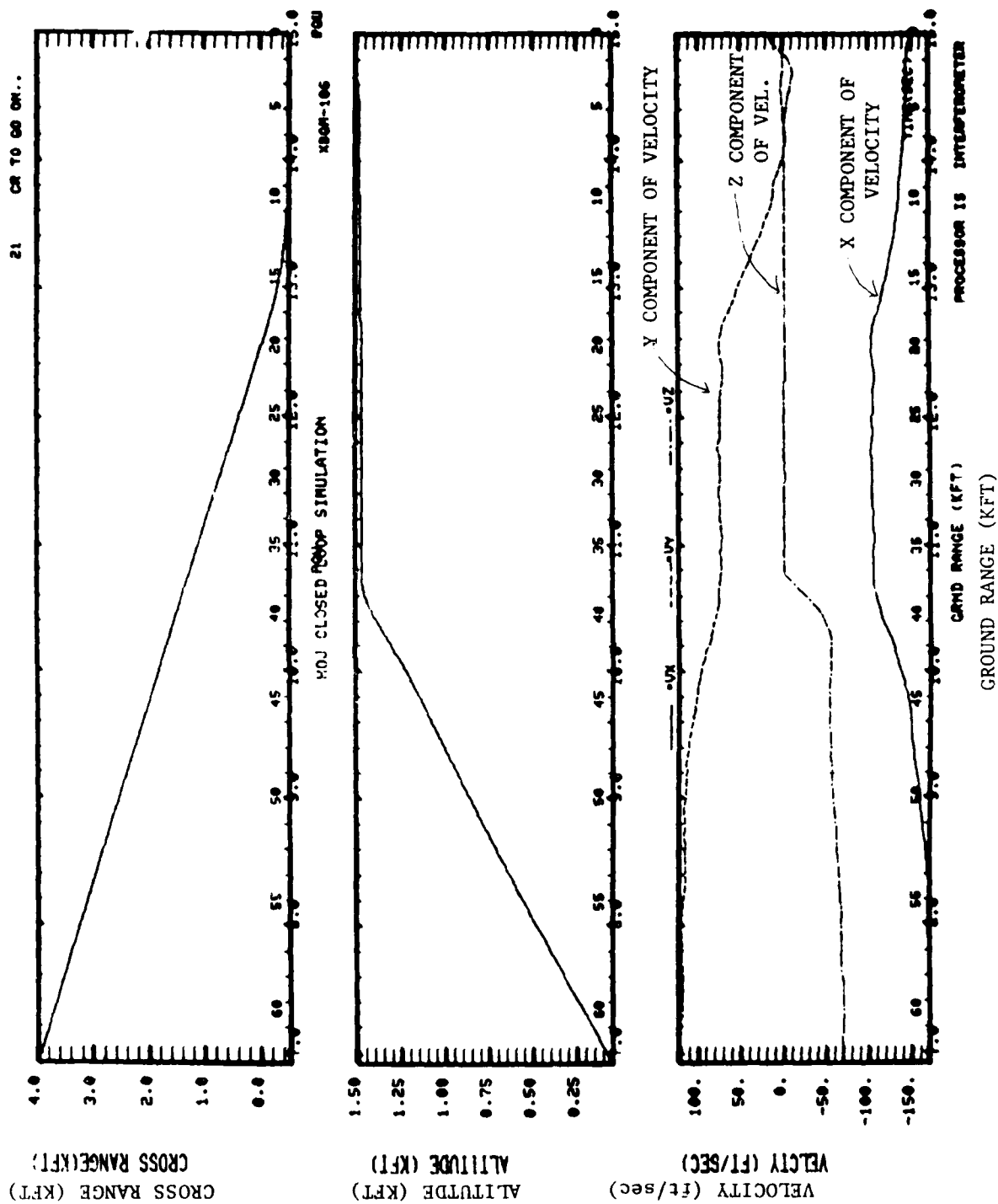
Fig. 7.22. Cross range as a function of ground range
Altitude as a function of ground range
Vehicle velocity as a function of ground range.

7-24

Appendix A
SUBROUTINES FOR INITIALIZATION MODULE

This appendix provides additional details relative to the subroutines used by the initialization module. It is most useful when used with the listings included with this report.

The following subroutines are included along with the page number on which they are found:

Subroutine PTIME (A,B,C,D,E,F,G)

Program PTIME

This routine obtains timing information. It is written in Assembler for the IBM 370. All the arguments correspond to values which are returned.

A = Integer, in milliseconds, which contains the amount of CPU time used by the task

B = Integer giving the hour of the day

C = Integer giving the minute of the hour

D = Integer giving the second of the minute

E = Integer giving the month

F = Integer giving the day of the month

G = Integer giving the year

Subroutine MISINT

Program MISINT

This routine initializes parameters LSTEST, LTHRUS as FALSE in labelled common TSTMIS for initializations in the flight dynamics simulation of the MAVERICK missile. This is included in this discussion for generality.

```
Subroutine READIT (6)
  Program $JCMLST
```

This subroutine reads the number which indexes the run and the title which labels the run. These quantities are specified in the EXEC. The EXEC input is also used to determine whether specular ground reflections will be modelled and initial conditions for the vehicle state vector (orientation, position, velocity). The following subroutines which are discussed below, are used to read information from the EXEC:

Subroutine RDRI

Subroutine YESNO

Subroutine WRTYN

Subroutine HOJRDT.

It should be noted that if ground reflections are not to be modelled variable IFGREF is 0; if they are to be modelled IFGREF is 1. The value of this parameter determines whether subroutines modelling specular ground reflections are to be called. IFGREF is a parameter in the subroutine READIT and is set by input from the EXEC which is read by subroutine YESNO.

```
Subroutine RDRI (I,J)
  Program RDRI
```

This subroutine reads the number used as a label to index the simulation run. It is **system dependent**.

I = 5 which corresponds to a read from the terminal

J = Index number input by the user

If J is input incorrectly, the subroutine returns REENTER until the input is correct.

A-3

```
Subroutine YESNO (TORF)
    Program YESNO
```

This subroutine reads the user response to the question of modelling
specular ground reflections. TORF is a logical variable which is set to TRUE
is the answer YES is given to the question and FALSE if the answer NO is
given. Either YES or NO must be input in response to this question otherwise
the subroutine will return the response REENTER. The program calls RDRC for
the reading of the input data from the EXEC.

```
Subroutine RDRC (I,A)
    Program RDRC
```

This subroutine reads A (either 'YES' or 'NO') from the EXEC. If A is
neither 'YES' or 'NO', the subroutine PUTCHR is called to output 'REENTER' to
the terminal. I is set to 5.

```
Subroutine PUTCHR (B)
    Program PUTCHR
```

This subroutine writes the work 'REENTER' to the terminal. B =
'REENTER'.

```
Subroutine WRTYN (6, TORF)
    Program WRTYN
```

This subroutine will write 'YES' to the terminal (unit 6) if specular
ground reflections are to be modelled, and 'NO' if they are not modelled de-
pending on the value of the logical variable TORF.

A-4

```
Subroutine HOJRDT (6)
  Program HOJRDT
```

This subroutine reads which receivers are to be considered for the simulation run. The capability exists to model vehicle performance in the specified tactical scenario, with a wide variety of candidate antenna/receiver/ signal processing systems for direction finding. It also accepts as input the initial position of the vehicle, and its speed and orientation angles with respect to Emitter 1. Given the orientation of the vehicle velocity vector relative to this emitter, the coordinates of the vehicle boresight with respect to an inertial coordinate system is determined.

OUTPUTS

LABELLED COMMON

HOJ1   PICK(I) is a logical array which speci-
                           fies which DF seekers are to be mod-
                           elled. Parameter I can vary from 1 to
                           5. PICK(2) = TRUE if and only if RDG
                           is specified in the EXEC as a receiver
                           option, and corresponds to the use of
                           the Enhanced Interferometer.

LABELLED COMMON

HOJ2   POSITION(3) = Missile position (ft)
                           VELCTY(3) = Missile velocity (ft/sec)
                           BOREST(3) = Missile boresight (ft)
                           These coordinates are defined relative
                           to an inertial system and are constantly
                           updated throughout the simulation run.

A-5

LABELLED COMMON

SPAHOJ   SPEED = Real missile speed (ft/sec)

          MISSAZ = Real initial azimuth planar
                   angle (degs) relative to Emit-
                   ter 1 (denoted by XMTRAZ)

          MISSEL = Real initial elevation planar
                   angle (degs) relative to Emit-
                   ter 1 (denoted by XMTRAZ)

LABELLED COMMON

HOJ4   INPOS(3) =   Initial missile position

          INTVEL(3) =   Initial missile velocity

          INTBOR(3) =   Initial missile boresight


```
Subroutine GETRCV (PICK, NUMRCV, PRINT, OUT, IN)
  Program HOJRDT
```

This subroutine determines which elements of the logical array PICK are TRUE depending on the receiver options which are specified. TDG must be specified in the EXEC for the enhanced interferometer to be modelled. When this character string is read by subroutine RDRCV, the second element in the logical array PICK is set to TRUE. Other character strings, when input in the EXEC, will cause other elements of the array PICK to be set to TRUE. If only TDG is specified all of the elements of PICK are set to FALSE except the second element [PICK(2)] which is set to TRUE.

INPUTS

In argument list   NUMRCV = 5

                    PRINT   = 6

                    OUT   = 6

                    IN   = 5

OUTPUTS

In argument list   PICK(I)   I = 1,5 determined as
noted above.

Subroutine RDRCV (IN, OUT, RCV, EOF)

  Program HOJRDR

This subroutine reads the characters in the EXEC file.

    IN = 5

    OUT = 6

    RCV = Character string specifying various receiver options

    EOF = Logical variable which is TRUE when character string END is
    read in the EXEC file.

Subroutine RDRXYZ (X,Y,Z)

  Program RDRXYZ

This subroutine reads in three real numbers, which specify the vehicle
position. It calls ERRSET, a system subroutine for checking input data for
errors.

Subroutine INTPOS (Speed, Az, El, Pos, Vel, Borest)

  Program GDDRONE

This subroutine uses initial specification of quantities Speed, Az, and
El to determine the vehicle boresight vector coordinates with respect to iner-
tial space. This subroutine calls INVERT.

```
Subroutine INVERT (POSX, VELP, VEL)
   Program GDDRONE
```

This subroutine uses information regarding the line-of-sight vector from missile to Emitter 1 (POSX) and knowledge of the velocity vector orientation relative to the line-of-sight (VELP) to determine velocity coordinates relative to inertial space (VEL), where POSX, VELP, and VEL are arrays.

```
Subroutine TITLPG
   Program CLSLOOP
```

This subroutine writes out some of the parameters associated with the emitters, such as position, wavelength, and polarization on the first page of output. On the second page of output, parameters associated with the multipath model are written out. For typical cases in which the number of rectangular and triangular surface elements, the number of buildings, and the number of aircraft is zero, and the number of emitters is 3, the following quantities are written out:

XMTR__X	Position of emitters
XMTR__Y	in ft.
XMTR__Z	
__VELX	Velocity of emitters
__VELY	in ft. (not incorporated
__VELZ	into the simulation model)
WL__	Wavelength of emitters
PL__	Polarization T = Vertical
	F = Horizontal
DIM__	Dimensions of transmit antenna (Irrelevant
	except that DIMDME must not be 0)

ISPGRD		0 if full integration is to be done
		1 if simple Fresnel reflection roughness co-efficient is to be computed
ERO		default complex dielectric constant
SH2O		default rms roughness height
NA	25	integer number of grid points along major and minor axis of ground
NB	11	reflection integration ellipse

Parameters for diffuse ground scattering are next written out. (It should be noted that diffuse scattering is not modelled in this iteration of $JCMLST.) Parameters for editing the various multipaths are also output to the terminal. These are also not used in the iteration of $JCMLST being presented. The quantities which are written out in TITLPG are all specified by the user in the block data file BDUHFM.

It should be noted that some of the calls in this program assume a particular graphics package. Subroutines PAGADV, ORTSPC, OHMP, OSHBD, and OSHDAC, which are called by TITLPG, also reside in program CLSLOOP.

Subroutine BXMWRT

Program BXMWRT

This subroutine writes out to the terminal blinking emitter parameters initial conditions of the vehicle and parameters associated with the seeker.

(1) For each emitter, the relative power, relative voltage, and parameters specifying the antenna pattern are written out.

(2) The receiver types that are being simulated are written out to the terminal. Since TDG is specified in the EXEC, the receiver that is modelled is the interferometer. Next the variables

A-9

$$FAD2 = \frac{d}{2} \text{ (meters)} \quad \text{Azimuth}$$

$$FED2 = \frac{d}{2} \text{ (meters)} \quad \text{Elevation}$$

$$FIA = 2\pi\frac{d}{\lambda} \text{ (dimensionless)} \quad \text{Azimuth}$$

$$FIE = 2\pi\frac{d}{\lambda} \text{ (dimensionless)} \quad \text{Elevation}$$

are output as well as the parameters DYNX, AM15, ITRK, IGAIN, and BWT. These parameters are used in the modelling of the enhanced interferometer. Lastly, initial vehicle parameters such as speed, misalignment angles in azimuth and elevation between the L.O.S. to Emitter 1 and the velocity vector, and initial position and velocity components are written.

(3) The number of blinking intervals and the off-time intervals for each emitter is written out.

```
Subroutine INIT
  Program INIT
```

This subroutine initializes the common areas of the signal in space routines based upon the input contained in the block data file BDUHFM.

```
Subroutine SURFSW (A,B)
  Program SURFSW
```

This subroutine modifies, if necessary, the SURFS array which contains the coordinates of the corners of the rectangles and triangles that are used to represent the ground plane. Other subroutines that are called expect that the corners will be in a specified order. This subroutine checks the SURFS

A-10

array, and, if necessary, modifies it to insure that the proper order is main-
tained. Integers A and B indicate the locations in the SURFS array. When the
number of rectangles and triangles is set to zero, as is typically the case,
this routine is not called.

```
Subroutine VSUB (A,B,C)
   Program VSUB
```

This subroutine subtracts vector B from vector A to give vector C, where
A, B, and C are arrays of order 3.

```
Subroutine NORML (A,B,C)
   Program NORML
```

This subroutine computes the cross product of vectors A and B and normal-
izes to produce the vector C.

```
Subroutine CROX (A,B,C)
   Program CROX
```

This subroutine computes the cross product of vectors and A and B and
puts the result in C.

```
Subroutine NoRMLZ (A)
   Program NORMLZ
```

This subroutine normalizes vector A.

```
Subroutine ORIENT (A,B,C)
  Program ORIENT
```

This subroutine computes the components of a vector relative to a vertical cylinder, given its components with respect to the Ath horizontal aircraft fuselage. This transformation is employed so that the same subroutines which model reflections off a horizontal aircraft fuselage can be used to model multipath off a vertical tail fin. This subroutine is typically not called for tactical scenarios and the number of aircraft is set to zero in block data.

```
Function DOT (A,B)
  Program DOT
```

This function computes the dot product of two vectors with coordinates A and B.

```
Subroutine ACTYPE (A,B,C,D,E,F,G,H)
  Program JCHOJ
```

This subroutine specifies the following parameters for certain aircraft

    A = Aircraft type
    B = Radius of fuselage
    C = Length of fuselage
    D = Radius of curvature of tail fin
    E = Length of tail fin
    F = Height of tail fin
    G = Total are of both wings
    H = Height of center of fuselage above the ground.

When the number of aircraft is set to zero this routine is not called.

```
Subroutine SCLHOJ
   Program SCLHOJ
```

This subroutine provides the scaling parameters used by the plotting programs.

```
Subroutine FLIGHT (STOP, NEWSEG)
   Program HOJRDT
```

This subroutine is used to determine when a particular receiver loop is to be ended and final miss distance calculated. The vehicle trajectory iteration is ended when one of several criteria are met:

(a) The index which specifies the flight time thru the Eq.

$$\text{FLIGHT TIME} = (\text{Index}) (\text{Drate})$$

exceeds a certain value, IND, which is specified in this subroutine

(b) The altitude of the vehicle is less than 0.1 ft.

(c) The altitude of the vehicle is less than the height of a particular emitter and the distance from the emitter is less than 50 ft.

If any of these criteria are met, logical variable STOP is set to .TRUE.. Logical variable NEWSEG is always set to .TRUE..

```
Entry XDIV (A,B)
   Program NORMLZ
```

This subroutine normalizes the receiver velocity vector by the speed of light.

```
Subroutine MINPOS (IXMTR)
   Program MINPOS
```

This subroutine determines the minimum miss distance relative to the emitter IXMTR.

```
Subroutine EMCASE (IXMTR, TEMP)
   Program CRSPLT1
```

This subroutine determines the coordinates of the vector TEMP from the vehicle to emitter IXMTR.

```
Subroutine UPDTMS (X, IXMTR, TEMP)
   Program UPDTMS
```

This subroutine determines the miss distance, X, from the vehicle to the emitter IXMTR using a point of closest approach analysis.

```
Subroutine MOVE (A,B)
   Program MOVE
```

Subroutine MOVE is used to load emitter parameters into labelled common XMTR.

A = arrays BLKA, BLKD, BLK2

B = arrays BLKT in common labelled XMTR

```
Subroutine INITR
   Program INITR
```

This subroutine computes the receiver position and velocity coordinates in the reference frame of equivalent vertical aircraft fuselages.

```
Subroutine ORIENT (A,B,C)
   Program ORIENT
```

This subroutine computes the components of a vector relative to a vertical cylinder given its components with respect to a horizontal cylindrical fuselage.

```
Subroutine INITXR
   Program INITXR
```

This subroutine computes the relative vector from the transmitter to the receiver, the wave number, and the wave number squared.

```
Function VNORMC (A,B,C)
   Program NVCRMC
```

This function finds the length of a vector with coordinates A,B,C

```
Function VNORM (A)
  Program VNORM
```

This function finds the length of a vector with coordinates in the three element array, A.

Appendix B

SUBROUTINES FOR SIGNAL-IN-SPACE MODULE

This appendix provides additional details relative to the subroutines used by the signal-in-space module and complements Section 3.0. It is most useful when used with the program listings which accompany this report. For convenience in using this appendix with Section 3.0, it is divided into four parts B.1, B.2, B.3, and B.4 which correspond with Sections 3.1 through 3.4.

B.1 Direct Wave Subroutines

The following are the subroutines related to the direct wave computation (discussed in Section 3.1) along with the page number on which they are found.

Subroutine	Page No.
CHKMLT	B-1
DIRWV	B-2
RCVANG	B-3
VNORMC	B-3
DOTC	B-3
ATAN22	B-4
SHDHMJ	B-4
CIRCSH	B-6
GZ	B-7
HMPROT	B-7
CSPR	B-7


```
Subroutine CHKMLT (A)
  Program HOJRDT
```

This subroutine checks the number of multipath components (A) and if A is greater than 147 stops execution of the program.

```
Subroutine DIRWV
  Program DIRWV
```

This subroutine determines the multipath parameters associated with the direct wave.

INPUTS

LABELLED COMMON RCVER	RCVR (3)	=	Receiver position relative to the transmitter (3-vector).
	RVEL (3)	=	Receiver velocity (3-vector) in inertial coordinates.
LABELLED COMMON	XMTR (3)	=	Transmitter position (3-vector).
	XVEL (3)	=	Transmitter velocity (3-vector).

OUTPUTS

LABELLED COMMON DIRWAV	DOPPD	=	Doppler of direct wave
	DOPPDR	=	Doppler of receiver (used in signal processor module to model the phase change in the multipath components over small increments of time).
	AZDIP	=	Azimuth planar angle specifying the location of the receiver with respect to the transmitter in the inertial reference frame.
	ELDIR	=	Elevation planar angle specifying location of receiver with respect to the transmitter in the inertial reference frame.
	AZIDIR	=	Azimuth planar angle relative to the vehicle boresight.
	ELIDIR	=	Elevation planar angle relative to the vehicle boresight.

B-2

```
Subroutine RCVANG (A,B,C,D,E,F,G,H)
   Program GDDRONE
```

This subroutine uses inertially referenced coordinates to determine azimuth and elevation angles relative to the vehicle. The arguments are

$$
\left.\begin{array}{l}
A = X \\
B = Y \\
C = Z
\end{array}\right\}
\begin{array}{l}
\text{vector coordinates to a} \\
\text{point in space in inertial} \\
\text{reference frame.}
\end{array}
$$

D,E,F   dummy arguments

G =   Planar azimuth angle relative to vehicle boresight

H =   Planar elevation angle relative to vehicle boresight.

```
Function VNORMC (A,B,C)
   Program VNORMC
```

This function computes the length of a three-component vector with A,B,C being the X,Y,Z coordinates.

```
Function DOTC (A,B,C,D,E,F)
   Program DOTC
```

This function computes the dot product of two three-component vectors with A,B,C being the coordinates of one vector and D,E,F the components of the other.

```
Function ATAN22 (A,B)
   Program ATAN22
```

This function computes the arc tangent of A/B and returns a value between $-\pi$ and π.

```
Subroutine SHDHMJ
   Program SHDHMJ
```

This subroutine models the shadowing effects due to circular cylindrical humps, and determines the associated multipath parameters. These parameters are like those found in DIRWV. Note that this subroutine is called only if IHUMP is set to 1 in block data file BDUHFM. The default value is zero.

KEY INPUTS

LABELLED COMMON

HUMPR	HUMPF (3) =	X,Y,Z coordinates of the front of circular cylindrical runway hump.
	HUMP (3) =	X,Y,Z coordinates of the middle of circular cylindrical runway hump.
	HUMPB (3) =	X,Y,Z coordinates of the back of circular cylindrical runway hump.

These coordinates assume the coordinate system shown in Fig. 3.4. With this coordinate convention, the coordinates HUMPF(2), HUMP(2), and HUMPB(2) are all zero.

LABELLED COMMON

XMTER	XMTR (3) =	X,Y,Z coordinates of the transmitter location.

B-4

LABELLED COMMON

RCVER   RCVR (3)   =   X,Y,Z coordinates of the receiver lo-
cation.

RELRCR (3) =   X,Y,Z coordinates of the vector from
receiver to the transmitter.

LABELLED COMMON

AIRPRT   RNWID   =   Runway width

LABELLED COMMON

DIRWAV   AZDIR   = $\left.\begin{array}{c}\\\\\\\end{array}\right\}$ Planar angles (Azimuth and Eleva-
tion) of the direct wave in the
                    ELDIR   =   transmitter coordinate system.

                    DOPPDR   =   Doppler of receiver

                    AZIDR   = $\left.\begin{array}{c}\\\\\\\end{array}\right\}$ Planar angles of the direct wave
as seen from the receiver in the
                    ELIDR   =   receiver coordinate system.

KEY OUTPUTS

LABELLED COMMON   TSHAD(1) <--> E_D/E_0 where E_D/E_0 is   ratio   of
diffracted wave to the free space sig-
nal.

                            AZSHAD(1) <--> AZDIR

                            ELSHAD(1) <--> ELDIR

                            AZISHAD(1)<--> AZIDR

                            ELISHAD(1)<--> ELIDR

                            RDOP(1)   <--> DOPPDR

The quantities TSHAD, AZSHAD, ELSHAD, AZISHAD, ELISHAD are analogous to the
multipath parameters determined in the call to DIRWV. The quantity TSHAD is 1
for the direct wave by definition.

LABELLED COMMON

SHCIRC   CENTER(2) =   Distance from the center of the hump to the vector from the receiver to the transmitter.

Subroutine CIRCSH (A,B,C,D)

  Program SHDHMJ

This subroutine carries out the detailed calculation of the shadowing effect due to a circular cylindrical hump.

A = Knife edge parameter (α)

B = α^2

C = Correction term to be subtracted from the knife edge result.

$$\left(\frac{G(X)}{u} e^{-j\alpha^2} \right)$$

D = Specifies the statement number in the calling subroutine that this subroutine will return to if certain criterion are met.

INPUTS

LABELLED COMMON

RCVER   RCVR (3)   =   Receiver position 3-vector in the inertial reference frame.

                             RELRCR (3) =   3-vector from the receiver to the transmitter in the inertial reference frame.

XMTER   XMTR (3)   =   Transmitter position 3-vector in the inertial reference frame.

k   =   Wave number

Function GZ(A)

  Program SHDHMJ

This is the complex function used to compute the shadowing effect due to the circular cylindrical hump.

Subroutine HMPROT (A,B,C,D)

  Program SHDHMJ

This subroutine transforms inertially referenced coordinates of a vector into a coordinate system based on the vector from the transmitter to the receiver.

A = cosine of angle of rotation

B = sine of angle of rotation

C = x, y, z of input coordinates

D = x, y, z of rotated coordinates

Subroutine CSPR (A,B,C)

  Program CSPR

This subroutine determines the Fresnel integral given argument C by calling IBM scientific subroutine CS.

A = Real part of Fresnel integral of C

B = Imaginary part of Fresnel integral of C

C = Argument of Fresnel integral

B.2 Ground Reflection Subroutines

The following subroutines related to the ground reflection calculation (discussed in Section 3.2), along with the page number on which they are located, are:

Subroutine CHKMLT (A)

Program HOJRDT

See Section B.1 above.

END
DATE
FILMED
6-83
DTIC

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

```
┌─────────────────────────┐
│ Subroutine GREFC        │
│   Program GREFC         │
└─────────────────────────┘
```

This subroutine determines ground reflection parameters if IFGREF is 1 and logical variable HUMP is false. The ground plane can be divided into 10 rectangular elements and 10 triangular elements, each with its own characteristic roughness coefficient (SH2S) and dielectric constant (ERS). The remaining regions are given default values of surface roughness (SH2O) and dielectric constant (ERO). Specular reflection parameters for each of these regions if ISPGRD is specified as 0 in block data BDUHFM. If ISPGRD is specified as 1 in the block data, a simplified method is used to compute the multipath parameters as noted in Section 3.2.

INPUTS

are described in GREFC comment statements.

OUTPUTS

ARGUMENT LIST

AMPL	=	Relative amplitude of ground reflection
PHAS	=	Relative phase of ground reflection
AZ	=	Planar azimuth in inertial XY plane
EL	=	Planar elevation angle in inertial XY plane
TDEL	=	Relative time delay of ground reflection
AZI	=	Planar azimuth receiver incidence angle
ELI	=	Planar elevation receiver incidence angle
DOP	=	Doppler shift of ground reflection
DOPR	=	Receiver component of Doppler shift

```
Subroutine VSUB (A,B,C)
  Program VSUB
```

This subroutine determines the difference between two vectors.

A = An array of three components for Vector 1.

B = An array of three components for Vector 2.

C = An array of three components of the vector difference between vectors 1 and 2.

```
Subroutine DTDEL (A,B,C,D)
  Program DTDEL
```

This subroutine determines the time delay of a specular ground reflection with respect to the direct wave.

A = Coordinates of the 3-component vector of the transmitter relative to specular point (in inertial space)

B = Coordinates of the 3-component vector of the receiver relative to the specular point (in inertial space)

C = Coordinates of the 3-component vector of the receiver relative to the transmitter (in inertial space)

D = Time delay of the specular ground reflection relative to the direct wave

```
Subroutine RCVANG (A,B,C,D,E,F,G,H)
  Program GDDRONE
```

See Section B.1 above.

```
Subroutine GRDROT (A,B,C)
   Program SPECGD
```

This subroutine transforms coordinates from the inertial system to that defined by the Ath surface element.

  A = Index which specifies which surface element is considered

  B = Original coordinates relative to inertial space

  C = Transformed coordinates relative to the Ath surface element

```
Subroutine GRDRTB (A,B,C)
   Program SPECGD
```

This subroutine transforms coordinates from the system defined by the Ath surface element of the inertial system.

  A = Index which specifies which surface element is considered

  B = Coordinates in system defined by the Ath surface element

  C = Coordinates relative to inertial coordinate system

```
Subroutine ROT (A,B,C,D)
   Program ROT
```

This subroutine applies a rotation of C radians about the D axis to the vector A.

  A = Original coordinates

  B = Transformed coordinates assuming rotation

  C = Size of angle (radians)

  D = Indicates which axis

```
Subroutine GRND (A,B,C)
  Program GRND
```

This subroutine determines which surface element contains a given point in the X-Y plane.

A = Coordinates of integration point

B = Number of the surface element which contains the point under consideration.

C = Indicates which surface elements are to be searched

```
Subroutine FRERFS (A,B)
  Program FRERFS
```

This subroutine computes ρ_s which is the product of the Fresnel reflection coefficient, R, and the attenuation factor, ρ_r, as opposed to using the Fresnel-Kirchoff diffraction formula and performing a detailed computation. This subroutine, as noted in Section 3.2, is called if ISPGRD is specified as 1 in block data file BDUHFM.

A = Direction vector from the transmitter to the specular point

B = Complex specular ground reflection coefficient ρ_s

INPUTS

LABELLED COMMON

XMTER	PLZ	=	True --> vertical polarization
		=	False --> horizontal polarization
	K	=	Wave number
	K2	=	k^2

LABELLED COMMON

SURFAC   ERG =   Complex dielectric constant of ground sur-
                        face

                SH2G =   Squared RMS roughness height of ground sur-
                        face

```
Function DOT (A,B)
  Program DOT
```

This function computes the dot product of two vectors A and B.

```
Function DDIST (A)
  Program DDIST
```

This function computes the length of a vector with coordinates A using double precision.

```
Function DOTC (A,B,C,D,E,F)
  Program DOTC
```

See Section B.1 above.

```
Function DREFC (A,B)
  Program DREFC
```

This subroutine calculates the differential reflectivity coefficient for a given grid point used in the approximation of the Fresnel integral.

A = 3-component vector giving the coordinates of the surface element

B = Index of surface element containing point A.

Subroutine SPECGD (A,B,C,D)
  Program SPECGD

This subroutine computes the location of the specular point for an arbitrary tilted ground region.

A = Location of specular point

$\left.\begin{array}{l} B = X'_{1n} \\ C = Y'_{1n} \\ D = X_{on} - X_{1n} \end{array}\right\}$ Parameters specifying the size size of the region over which the Fresnel integration is to be performed.

KEY INPUTS
LABELLED COMMON

GRDRT	RECTA =	X,Y coordinates of fourth corner of I-th rectangle (found in subroutine INIT)
	RTCRS =	Orientation vector for I-th rectangle or triangle (found in subroutine INIT)
RTSPEC	MRTSPC =	Indices to be used in computing specular ground reflection for rectangles and triangles (found in subroutine INIT)
	NSPEC =	Number of specular ground reflection components (found in subroutine INIT)
GRDTRN	DICS =	9 direction cosines to specify the orientation of the surface elements
	XYZOG =	Coordinates to translate origin for each surface element considered

	SLOPE	= Slopes of sides of triangles
	YINT	= Y intercept of sides of each triangle
XMTER	XMTR	= Transmitter position
	XVEL	= Transmitter velocity
RCVER	RCVR	= Receiver position
SURFACE	NS	= Number of surface elements (less than or equal to 20)

Subroutine CHSPEC (A,B,C)

  Program SPECGD

This subroutine repositions the specular point for the Ath surface element

A = Index to specify which surface element is being considered.

B = Coordinates of specular point.

C = Coordinates of repositioned specular point.

INPUTS

LABELLED COMMON

SURFAC	NT	= Number of triangles
	NR	= Number of rectangles
	SURFS	= Defining coordinates of the surface elements

```
Subroutine WGTAMP (A,B,C,D,E)
  Program MLTPLOT
```

This subroutine applies amplitude weighting due to the transmitter an-
tenna pattern.

A = Multiple emitter index

B = Adjusted multipath amplitude normalized with
    respect to that of the direct wave from the azi-
    muth transmitter

C = Multipath azimuth angle at the transmitter

D = Multipath elevation angle at the transmitter

E = Raw multipath amplitude

KEY INPUTS

LABELLED COMMON

AZBLNK   NAZBLK, TAZBLK

DMBLNK   NDMBLK, TDMBLK

EIBLNK   NEIBLK, TEIBLK

E2BLNK   NE2BLK, NE2BLK

         where   N__BLK = Number of off-time intervals

                 T__BLK = Off-time intervals

DIRWAV   AZDIR = Planar azimuth angle of direct wave

                ELDIR = Planar elevation angle of direct wave

CMFLTP   TIME   = Time along flight path

BXMPAR   KXMTRK = Tracking option parameter

                BAZTRK = Planar azimuth pointing angle

                BELTRK = Planar elevation pointing angle

B-16

```
Function GXMTR (A,B,C)
   Program JCHOJ
```

This subroutine is used to determine the gain of the Ath transmitter given azimuth and elevation planar angles of B and C. This subroutine is called twice. The first time modifications to the specular reflection parameters are determined. The second time adjustments are made to the direct wave.

```
Subroutine SETVAL (A,B,C,D,E,F,G,H,I,J
   Program SETVAL
```

This subroutine is called to transfer parameters associated with direct wave and specular reflection to labelled common areas MULTI and MULTIP.

A = Index indicating where quantities are to be loaded in labelled common arrays.

B = Amplitude

C = Phase

D = Azimuth planar angle relative to the transmitter

E = Elevation planar angle relative to the transmitter

F = Time delay

G = Magnitude of the Doppler shift

H = Receiver component of the Doppler shift

I = Azimuth planar angle relative to receiver

J = Elevation planar angle relative to receiver

B-17

```
Subroutine DIGLOD (A,B,C,D,E,F,G,H,I,J)
   Program MLTPLOT
```

This subroutine is called to transfer parameters associated with the direct wave and specular reflection to the labelled common area DIAGP.

A = Index indicating which quantitites are to be used in labelled common MULTI.

B = Signal processing index (2 for the enhanced interferometer)

C = Multiple emitter index

D = Index specifying which multipath obstacle is under consideration (1 for the direct wave, 2 for the ground, 3 through 12 for buildings, 13 through 22 for aircraft).

E = Weighted multipath amplitude normalized with respect to that of the direct wave from emitter 1.

F = Raw amplitude

G = Multipath planar azimuth angle relative to emitter axes

H = Multipath planar elevation angle relative to emitter axes

I = Receiver fractional Doppler frequency

J = Time delay

```
Subroutine GAIN (A,B,C,D,E)
  Program CCGAIND
```

This subroutine characterizes the receiver antenna pattern:

A = Index of multipath component

B = Antenna element

C = Emitter number

D = Modified phase

E = Modified gain

B.3 Building Reflection Subroutines

The following subroutines related to building reflection calculations (discussed in section 3.3), along with the page number on which they are located, are:

Subroutine	Page Number
CHKMLT	B-20
BREFC	B-20
BCOORD	B-20
BCOMP	B-21
CSPR	B-21
VSUB	B-21
DTDEL	B-21
BCOORT	B-22
POLZN	B-22
RCVANG	B-23
VNORM	B-23
VNORMC	B-23
WGTAMP	B-23
GXMTR	B-23
SETVAL	B-24
DIGLOD	B-24
GAIN	B-24

```
Subroutine CHKMLT (A)
  Program ''CJRDT
```

See Section B.1 above.

```
Subroutine BREFC (A,B,C,D,E,F,G,H,I,J,K)
  Program BREFC
```

This subroutine determines the reflection multipath for a given building.

A = Index specifying which building is being considered

B = Index specifying which path is being considered

C = Relative amplitude of the building reflection

D = Phase of the building reflection

E = Planar azimuth angle

F = Planar elevation angle

G = Relative time delay of the building reflection

H = Doppler shift of the building reflection

I = Receiver component of building reflection Doppler shift

J = Planar incident azimuth angle

K = Planar incident elevation angle

```
Subroutine BCOORD (A,B,C)
  Program BCOORD
```

This subroutine converts ground position coordinates to a building centered coordinate system.

A = Building number

B = Ground coordinates of position vector

C = Building centered coordinates of position vector

B-20

```
Entry BCOMP (A,B,C)

  Program BCOORD
```

This subroutine converts velocity vectors in a ground system to a building referenced system.

A = Building number

B = Ground coordinates of velocity vector

C = Building centered coordinates of velocity vector

```
Subroutine CSPR (A,B,C)

  Program CSPR
```

Fresnel integral subroutine. See Section B.1.

```
Subroutine VSUB (A,B,C)

  Program VSUB
```

Vector subtraction subroutine. See Section B.2

```
Subroutine DTDEL (A,B,C,D)

  Program DTDEL
```

Subroutine for determining time delay associated with a building multipath relative to the direct wave. See Section B.2

```
Subroutine BCOORT (A,B,C)
  Program BCOORT
```

This subroutine converts building referenced coordinates of a position vector to a ground referenced coordina ° system.

A = Building number

B = Building coordinates of position vector

C = Ground coordinates of position vector

```
Subroutine POLZN (A,B,C,D,E,F,G)
  Program POLZN
```

This subroutine computes the equivalent Fresnel reflection coefficient for reflection from a tilted planar facet.

A = Relative complex dielectric constant

B = Ground unit normal vector (not used in calculations)

C = Planar facet unit normal vector

D = Vector from specular point to the transmitter

E = Vector from specular point to the receiver

F = 1 for a perfect conductor; 0 otherwise

G = Equivalent Fresnel reflection coefficient

KEY INPUT

LABELLED COMMON

XMTR   PLZ =   Logical variable describing emitter polarization; true for vertical polarization, false for horizontal polarization.

```
Subroutine RCVANG (A,B,C,D,E,F,G,H)
   Program GDDRONE
```

See Section B.1.

```
Function VNORM (A)
   Program VNORM
```

See Appendix A.

```
Function VNORMC (A,B,C)
   Program VNORMC
```

See Section B.1.

```
Subroutine WGTAMP (A,B,C,D,E,F)
   Program MLTPLOT
```

This subroutine modifies the specular reflection parameters of a building depending on the transmitter antenna pattern.   See Section B.2.

```
Subroutine GXMTR (A,B,C)
   Program JCHOJ
```

See Section B.2

```
Subroutine SETVAL (A,B,C,D,E,F,G,H,I,J)
   Program SETVAL
```

See Section B.2.

```
Subroutine DIGLOD (A,B,C,D,E,F,G,H,I,J)
   Program MLTPLOT
```

See Section B.2.

```
Subroutine GAIN (A,B,C,D,E)
   Program CCGAIND
```

See Section B.2.

B.4 Aircraft Reflection Subroutines

The following subroutines related to building reflection calculations (discussed in Section 3.4), along with the page number on which they are found, are:

Subroutine	Page No.
CHKMLT	B-25
FREFC	B-25
ACTYPE	B-26
ORIENT	B-27
CYLREF	B-27
VSUB	B-28
XDIV	B-28

```
Subroutine CHKMLT (A)
  Program HOJRDT
```

See Section B.1 above.

```
Subroutine FREFC (A,B,C,D,E,F,G,H,I,J,K)
  Program FREFC
```

This subroutine determines the multipath quantities associated with re-flections off aircraft fuselages.

A = Airplane number

B = Multipath component number

C = Relative amplitude of fuselage reflection

B-25

D = Relative phase of fuselage reflection

E = Planar azimuth angle with respect to the trans-
    mitter

F = Planar elevation angle with respect to the
    transmitter

G = Relative time delay of fuselage reflection

H = Doppler shift of fuselage reflection

I = Receiver component of Doppler shift

J = Planar azimuth angle with respect to receiver

K = Planar elevation angle with respect to receiver

Subroutine ACTYPE (A,B,C,D,E,F,G,H)

Program JCHOJ

This subroutine sets the parameters for specified aircraft which charac-
terizes their geometry. These quantities are presently specified for aircraft
such as the DC-10 and the 707 because the program was originally designed to
study multipath environments in the vicinity of airfields.

A = Aircraft type

B = Fuselage radius

C = Fuselage length

D = Tail fin radius of curvature

E = Length of tail fin

F = Height of tail fin

G = Total area of both wings

H = Height of center of fuselage above ground

```
Subroutine ORIENT (A,B,C)
   Program ORIENT
```

This subroutine computes the equivalent vertical cylindrical components of a vector given its ground components with respect to a horizontal cylindrical fuselage.

A = Number specifying which airplane fuselage is under consideration.

B = Ground components of vector with respect to a horizontal cylinder

C = Vertical cylindrical components

```
Subroutine CYLREF (A,B,C,D,E,F,G,H,I,J,K,L)
   Program CYLREF
```

This subroutine computes the multipath parameters of a smooth, perfectly-conducting section of a vertical cylindrical surface.

A = Index to specify which airplane is under consideration

B = Index to specify which multipath component is under consideration

C = Reflection amplitude

D = Reflection phase

E = Planar azimuth angle with respect to inertial X axis

F = Planar elevation angle with respect to inertial X axis

G = Relative time delay

H = Doppler shift

I = Receiver component of Doppler shift

J = Specifies which vertical direction represents true vertical (so as to use the same program for modelling reflections from horizontal fuselages and vertical tail fins)

K = Planar incident azimuth angle

L = Planar incident elevation angle

Subroutine VSUB (A,B,C)

  Program VSUB

See Section B.2.

Subroutine XDIV (A,B)

  Program NORMLZ

This is the entry point to subroutine NORMLZ used to normalize a vector.

A = Coordinates of vector

B = Length of vector

Subroutine NORMLZ (A)

  Program NORMLZ

This subroutine normalizes a vector.

A = Coordinates of vector.

```
Subroutine ROTATE (A,B,C)
  Program ROTATE
```

This subroutine finds the components of a dimensional vector in a rotated coordinate system.

A = Coordinates of vector in unrotated coordinate system

B = Coordinates of vector in rotated coordinate system

C = Coordinates of unit vector in rotated x direction relative to unrotated system.

```
Subroutine CSPR (A,B,C)
  Program CSPR
```

See Section B.1.

```
Subroutine ORIEND (A,B,C)
  Program POLZN
```

This subroutine transforms the vector coordinates from the cylinder coordinate system to the inertially referenced system.

A = Airplane number

B = Vector in fuselage coordinate system

C = Vector in ground coordinate system

```
Subroutine POLZN (A,B,C,D,E,F,G)
   Program POLZN
```

See Section B.3.

```
Subroutine TREFC (A,B,C,D,E,F,G,H,I,J,K)
   Program TREFC
```

This subroutine performs the same function for tail fins that FREFC does for fuselage elements.

```
Subroutine WGTAMP
```

See Section B.3.

```
Subroutine GXMTR
```

See Section B.3.

```
Subroutine SETVAL
```

See Section B.3.

```
┌─────────────────────────────────┐
│                                 │
│  Subroutine DIGLOD              │
│                                 │
│                                 │
└─────────────────────────────────┘
```

See Section B.3.

```
┌─────────────────────────────────┐
│                                 │
│  Subroutine GAIN                │
│                                 │
│                                 │
└─────────────────────────────────┘
```

See Section B.3.

```
┌─────────────────────────────────┐
│                                 │
│  Subroutine MAXMLT (A)          │
│    Program MAXMLT               │
│                                 │
└─────────────────────────────────┘
```

This subroutine determines the scaling parameters for multipath diagnostics (plots) at a particular evaluation point. The quantities determined are loaded into labelled common DIGSCL.

```
┌─────────────────────────────────┐
│                                 │
│  Subroutine PTIME (A,B,C,D,E,F,G)│
│    Program PTIME                │
│                                 │
└─────────────────────────────────┘
```

This subroutine is used to determine the CPU time required for multipath calculations. See Section 2.0 and Appendix A.

```
┌─────────────────────────────────┐
│                                 │
│  Subroutine DIGOUT (A,B)        │
│    Program MLTPLOT              │
│                                 │
└─────────────────────────────────┘
```

This subroutine writes out multipath data to disk (UNIT 14).

Appendix C

SUBROUTINES FOR SEEKER MODULE

This appendix provides additional details of the subroutines used for the seeker calculations. It is most useful when it is used with the listings included with this report.

The following subroutines are included along with the page number of which they are found.

C.1 GENERAL REQUIREMENTS AND DEFINITIONS

Requirements for Interferometer Signal Processing to be Used

1. Specify TDG in response to question on signal processing option to be used in exec TESTSIM2.

2. Load CAT1, CSRPVTRK, C1HOJG2, and CCGAIND in that order in the load sequence

Quantities Required for Signal Processing Computations

The following quantities are to be specified by the user in the block data file BDUHFM:

FAD2 = (Azimuth) d/2   (Meters)

FED2 = (Elevation) d/2   (Meters

FlA = (Azimuth) $2\pi d/\lambda$ (Dimensionless)

FlE = (Elevation $2\pi d/\lambda$ (Dimensionless)

where d is the spacing between antenna elements, λ is the wavelength of the incident radiation

DYNX = Scaling parameter X in the relationship

$$\theta = X(PHIP) \quad (Default = 1)$$

AMI5 = Quantity used for resolving phase difference (Default = 0)

ITRK = 1 for tracker implementation

IGAIN = 1 for modelling of seeker antenna characteristics

BWT = Coefficient used in some version of the subroutine modelling antenna gain (although not in CCGAIND).

DRATE = *Simulation interval for determining signal in space parameters, determining planar angle estimates, and updating autopilot (nominally 0.1 sec. for this vehicle)*

NXMTR = Number of emitters (less than or equal to 4)

The following quantities are determined in the multipath computation block:

AMP = multipath amplitude

PHASE = multipath phase

TAU = multipath time delay

RDOP = multipath fractional doppler frequencies

AZP = multipath azimuth incidence angles relative to the receiver

ELP = multipath elevation incidence angles relative to the receiver

OMEG = frequency of emitter signal

C.2 SEEKER MODULE SUBROUTINES

```
Subroutine HOJ (IANT, MISSLE)
   Program HOJRDT
```

This subroutine is the driver which determines the planar angles that specify the orientation of the estimated line-of-sight vector with respect to the vehicle boresight. When the enhanced interferometer is to be invoked then the parameter MISSILE is 2 and the parameter IANTI is set to 1 in the overall simulation driver, JCMLST.

```
Subroutine RDPHAS (IY)
   Program HOJRDT
```

Specifying TDG in the EXEC file TESTSIM2 as noted in A.1 calls this subroutine which adds a random phase angle between 0 and 2π to the multipath phase variable phase contained in labelled common MULTI. This subroutine calls a system subroutine for generating the random number IY.

```
Subroutine DLS (ITH)
   Program CAT1
```

As for RDPHAS, specifying TDG in the EXEC calls this subroutine which calls, in turn, the subroutine C1HOJ for the enhanced interferometer. This subroutine generates the planar angles by modelling the antenna and signal processor and the tracker if it is invoked.

Note that RDPHAS and DLS are called twice. On the first call, the azimuth planar angle is calculated and on the second, the elevation angle. The following array contains the angles:

HOJERR(1) = Azimuth planar angle

HOJERR(2) = Not used

HOJERR(3) = Elevation planar angle

This subroutine also generates planar angles between the vehicle velocity vector, boresight vector, and seeker estimated LOS relative to the true line-of-sight to the emitters in the scenario that has been specified.

```
Subroutine C1HOJ (IP)
   Program C1HOJG2
```

This subroutine models the antenna pattern, signal and angle processing. The parameter IP is set to 1 in the calling subroutine DLS. Details of this subroutine are given in Section 4.2.

```
Subroutine GAIN (A,B,C,D,E)
   Program CCGAIND
```

This subroutine models the specific characteristics of the EHT antenna given the angle of arrival of the multipath signals. The parameters passed are:

A = Index of the multipath signal

B = Antenna element

C = Index of the emitter

D = Phase change

E = Antenna gain

```
Subroutine RESOLV (A,B,C,D)
   Program C1HOJG2
```

This subroutine resolves angle A into the angular region defined by angles B and C. The output of the subroutine is angle D.

```
Subroutine VSUB (A,B,C)
   Program VSUB
```

This subroutine computes the vector difference between A and B giving the result in C.

```
Subroutine CONVER (A,B,C)
   Program C1HOJG2
```

This subroutine converts the planar azimuth and elevation angles associated with the Ath multipath component to the conical angles B and C.

```
Subroutine ERRMOD (A,B)
   Program ERRMOD
```

This subroutine reads input defining the magnitude and characteristics of error sources and perturbs the angles determined by the enhanced interferometer processor. Gaussian noise corrupting the angles is modelled as well as constant biases and blind range effects as well as errors associated with the deviation from the ideal interferometer relationship. These errors are based on measurements. The parameters are defined as follows:

A = Perturbed azimuth angle (radians)

B = Perturbed elevation angle (radians)

Subroutine EMTLOD (A,B,C,D)

Program CRSPLT1

This subroutine loads the x,y,z coordinates of the Ath emitter into arguments B, C, and D.

Subroutine UPDTMS (A,B,C)

Program UPDTMS

This subroutine determines the miss distance, A, from the Bth emitter given the vector C from the vehicle to that emitter.

Subroutine ATRPLT (A,B,C,D,E)

Program GDDRONE

This subroutine interpolates tables A and B to find E for a given D. These tables have C element.

Subroutine ATRPL2 (A,B,C,D,E,F,G,H)

Program GDDRONE

This subroutine performs a two dimensional interpolation using tables A and B with dimensions D and E, respectively, by looking up values in table C. Arguments F and G are used for determination of the proper indices. The output is H.

```
Subroutine TRACK (A,B)
    Program CSRPVTRK
```

This subroutine, which is described in some detail in Section 4.3, takes planar azimuth and elevation angles A and B and replaces them if necessary with revised values based upon several criteria.

```
Subroutine EMCASE (A,B)
    Program CRSPLT1
```

This subroutine determines the vector B from the vehicle to the Ath emitter.

```
Subroutine RCVANG (A,B,C,D,E,F,G,H,)
    Program GDDRONE
```

This subroutine determines azimuth and elevation planar angles in the receiver coordinate system given the inertially referenced coordinates of the vector from receiver to target.

A = X $\left.\vphantom{\begin{array}{c}X\\Y\\Z\end{array}}\right\}$ vector coordinates to a point in
B = Y space in inertial reference frame
C = Z

D = dummy

E = dummy

F = dummy

G = planar azimuth angle relative to vehicle boresight

H = planar elevation angle relative to vehicle boresight

```
Subroutine INVERT (A,B,C)
   Program GDDRONE
```

This subroutine determines the vector coordinates, C, in the inertial frame using vector coordinates, B, in the body frame and coordinates of the boresight vector A in the inertial frame.

```
Subroutine ATRPLT (A,B,C,D,E)
   Program ERRMOD
```

This subroutine interpolates in tables A and B to find E for a given C. The tables have C elements.

```
Subroutine PUTHOJ (A,B)
   Program CRSPLT2
```

This subroutine writes out the vehicle position, velocity and boresight coordinates, planar angle estimates from the tracker, error angles and planar difference angles to the disk.

Appendix D

SUBROUTINES FOR FLIGHT DYNAMICS MODULE

This appendix provides additional details relative to the subroutines used by the flight dynamics module. It is most useful when used with the listings included with this report.

The following are the subroutines included along with the page number on which they are found.

```
Subroutine FLTDYN (A)

  Program GDDRONE
```

This subroutine is the driver which controls the sequencing of all of the subroutines in the flight dynamics module. When the enhanced interferometer is used (which is the case for the description given in this report), the parameter A is set to 2.

<u>INPUTS</u>

LABELLED COMMON

FLTIND   INDEX   =   Integer index which specifies how many interations have been made in the multipath computations.

AIRPRT   DRATE   =   Interval over which multipath quantities are determined.

<u>OUTPUTS</u>

LABELLED COMMON

SAMHLD   INTRTE   =   Interval for performing integration of the equations of motion.

                        TWD   =   } 1. constants associated with the

                        WAV   =   } 1. tunable integration scheme

This subroutine is called by the master driver for the simulation $JCMLST.

| Subroutine CRSGID (PTCHCM, YAWCM) |
| Program GDGUID |

This is the guidance computer subroutine which processes the planar azimuth and elevation angles to produce pitch and yaw commands for the autopilot. The inputs and outputs of the subroutine are:

<u>INPUTS</u>

LABELLED COMMON

HOJ3   HOJERR(1) =   Planar azimuth angle (rads)

                 HOJERR(3) =   Planar elevation angle (rads)

| GYRFD | OAQ | = | Filtered value of corrupted pitch and body rate (deg/sec) |
| | AOR | = | Filtered value of corrupted yaw body rate (deg/sec) |

OUTPUTS

SUBROUTINE ARGUMENTS

| | PTCHCM | = | Pitch steering command (deg/sec) |
| | YAWCM | = | Yaw steering command (deg/sec) |

This subroutine is called by FLTDYN the driver program for the flight dynamics part of the simulation. It is called every DRATE seconds.

Subroutine AUTPLT (PTCHCM, YAWCM, DELEN, DELRN, AILDEG)

Program GDDRONE

This is the autopilot subroutine which accepts commands from the guidance computer to produce elevator, rudder, and aileron positions. The inputs and outputs are:

INPUTS

| SUBROUTINE ARGUMENTS | PTCHCM | = | Guidance pitch command |
| | YAWCM | = | Guidance yaw command |

LABELLED COMMON

| GYRFD | OAQ1 | = | Corrupted pitch rate measurement (after modelling of the gyroscope errors) |
| | OAR1 | = | Corrupted yaw rate measurement (after modelling of the gyroscope errors) |

HOJ2	POSITN(3)	=	Perfect elevation measurement
DATRNS	PSIN	=	Perfect yaw measurement
XANROT	FIRF	=	Perfect roll measurement

OUTPUTS

SUBROUTINE ARGUMENTS	DELEN	=	Elevator deflection (degs)
	DELRN	=	Rudder deflection (degs)
	AILDEG	=	Aileron deflection (degs)

This subroutine is called by FLTDYN the driver program for the flight dynamics part of the simulation every INTRTE seconds.

Subroutine AERDYN (RALPHA, RBETA, DELEN, DELRN, AILDEG, ALAPRAT,

THREXT, CXN, CYN, CZN, COEFL, COEFM, COEFN, VWXN, VWYN, VWZN, SMP,

SMQ, SMR)

Program GDDRONE

This is the aerodynamics subroutine which provides the aerodynamic forces for the equations-of-motion. The inputs and outputs are:

INPUTS

SUBROUTINE ARGUMENTS	RALPHA	=	Angle of attack (rads)
	RBETA	=	Sideslip (rads)
	DELEN	=	δ_E (degs)
	DELRN	=	δ_R (degs)
	AILDEG	=	δ_A (degs)
	ALPRAT	=	Rate of change of α (rads/sec)

D-4

$$\left.\begin{array}{l} \text{VWXN} \\ \text{VWYN} \\ \text{VWZN} \\ \text{SMP} \\ \text{SMQ} \\ \text{SMR} \end{array}\right\}$$ Linear and angular velocities associated with the wind field

LABELLED COMMON

SWMARK	PMOD1	=	Logical variable to indicate pitchover (.TRUE. implies the vehicle is in the dive)
DATRNS	DELRNP	=	δ_R (rads)
	DELENP	=	δ_E (rads)
HOJ2	POSITN(3)	=	Vehicle altitude
AFRDBK	OAQN	=	Q (deg/sec)
	OARN	=	R (deg/sec)
	OAPN	=	P (deg/sec)

OUTPUTS

$$\left.\begin{array}{l} \text{CXBN} = C_x \\ \text{CYN} = C_y \\ \text{CZN} = C_z \\ \text{COEFL} = C_L \\ \text{COEFM} = C_M \\ \text{COEFN} = C_N \end{array}\right\}$$ Aerodynamic Coefficients

THREXT = Thrust

This subroutine is called by FLTDYN the driver program for the flight dynamics part of the simulation every INTRTE seconds.

```
Subroutine EQUAMO (THREXT, CXN, CYN, CZN, COEFL, COEFN, VELP,
RALPHA, RBETA, ALPRAT, VWXN, VWYN, VWZN, SMP, SMQ, SMR)
Program GDDRONE
```

This is the equations-of-motion subroutine which computes the accelera-
tions and velocities of the vehicle in its own axis system, along with the
space orientation. In addition, the velocities of the wind field relative to
the body axes and corruption of the body angular rates by gyro errors are also
computed. The inputs and outputs are:

INPUTS

SUBROUTINE ARGUMENTS   THREXT   = Thrust

                              CXN   = C_x

                              CYN   = C_y

                              CZN   = C_z

                              COEFL   = C_L

                              COEFM   = C_M

                              COEFN   = C_N

OUTPUTS

LABELLED COMMON

   ARFDBK   OAQN   = Q (degs/sec)

                              OARN   = R (degs/sec)

                              OAPN   = P (degs/sec)

                              VUN   = V_X; velocity along the body x
                                         axis (ft/sec)

                              VVN   = V_Y; velocity along the body y
                                         axis (ft/sec)

                              VWN   = V_Z; velocity along the body z
                                         axis (ft/sec)

                              D-6
```

ARGUMENT LIST       VELP(3)   =    ⎤   VELP(1)   =    $V_X$
                                    ⎬   VELP(2)   =    $V_Y$
                                    ⎦   VELP(3)   =    $V_Z$

                    RALPHA    =    $\alpha$ (rads/sec)

                    RBETA     =    $\beta$ (rads)

                    ALPRAT    =    $\dot{\alpha}$ (rads/sec)  ⎤  linear and
                    VWXN      =    $V_{wx}$ (ft/sec)          │  angular
                    VWYN      =    $V_{wy}$ (ft/sec)          │  velocities
                    VWZN      =    $V_{wz}$ (ft/sec)          ⎬  determined in
                    SMP       =    $P_w$ (rads/sec)           │  the call to
                    SMQ       =    $Q_w$ (rads/sec)           │  subroutine
                    SMR       =    $R_w$ (rads/sec)           ⎦  WIND


LABELLED COMMON

  DATRNS            ATACKP    =    $\alpha$ (degs)

                    NPN       =    Pitch load factor (ft/sec$^2$)

                    ATACKY    =    $\beta$ (degs)

                    NYN       =    Yaw load factor (ft/sec$^2$)

                    AQNP      =    Q (rads/sec)

                    THETN     =    $\Theta$ (rads)

                    ARNP      =    R (rads/sec)

                    PSIN      =    $\psi$ (rads)


  HOJ2              BOREST(3) =         Projection of unit vector
                                        pointed along vehicle bore-
                                        sight on inertial X,Y,Z axes


  ROLPLT            FIDOT     =    $\dot{\Phi}$ (rads/sec)

                    FI        =    $\Phi$ (rads where $-\infty \leq \Phi \leq \infty$)


  XANROT            FIRF      =    $\Phi$ (rads where $0 \leq \Phi < 2\pi$)


D-7

| GYRFD | OAQ | = | $F\{Q'\}$ (°/sec) | Filtered value of Q |
| | OAQ1 | = | $Q'$ (°/sec) | pitch rate corrupted by measurement errors |
| | OAR | = | $F\{R'\}$ (°/sec) | Filtered value of R' |
| | PAR1 | = | $R'$ (°/sec) | yaw rate corrupted by measurement errors determined by the call to GYERR. |

This subroutine is called from FLTDYN every INTRTE seconds.

---

Subroutine KINEM (VELP)

Program GDDRONE

---

This kinematics subroutine takes linear velocities in the vehicle frame and transforms them to linear velocities in the inertial frame using the Euler angle defined matrix. These velocities are integrated to determine the updated vehicle position in space. The inputs and outputs are:

INPUTS

ARGUMENT LIST

| VELP(3) | VELP(1) | = | $V_X$ | velocities in |
| | VELP(2) | = | $V_Y$ | vehicle reference |
| | VELP(3) | = | $V_Z$ | frame |

OUTPUTS

LABELLED COMMON

| HOJ2 | POSITN(3) | = | 3-component vector describing position in inertial space |

$$\text{VELCTY}(3) = \text{3-component vector describing velocity in interial space}$$

$$\text{DATRNS} \qquad \text{VN} = \sqrt{\sum_{J=1}^{3} \text{VELCTY}^2 (I)}$$

$$\text{GAMPN} = \tan^{-1}\left(\frac{\text{VELCTY}(3)}{\sqrt{\text{VELCTY}^2(1) + \text{VELCTY}^2(2)}}\right)$$

$$\text{GAMYN} = \tan^{-1}\left(\frac{\text{VELCTY}(2)}{\text{VELCTY}(1)}\right)$$

This subroutine is called from FLTDYN every INRTE seconds.

---

Subroutine WIND (A,B,C,D,E,F)

Program GDDRONE

---

This subroutine provides the wind linear and angular velocities in the vehicle frame of reference. The arguments are:

$$A = V_{wx} \text{ (ft/sec)} \quad \left.\begin{array}{l} \text{linear and} \end{array}\right.$$
$$B = V_{wy} \text{ (ft/sec)} \quad \text{angular}$$
$$C = V_{wz} \text{ (ft/sec)} \quad \text{velocities}$$
$$D = P_w \text{ (rads/sec)} \quad \text{associated}$$
$$E = Q_w \text{ (rads/sec)} \quad \text{with the wind}$$
$$F = R_w \text{ (rads/sec)} \quad \text{field}$$

It is first called from FLTDYN for initialization and then it is called from EQAMO every INTRTE seconds.

```
Subroutine RCWIND (A,B,C,D,E,F)
  Program GDDRONE
```

This subroutine transforms wind field velocities from inertially refer-
enced coordinates to vehicle referenced coordinates. The arguments are

A = X inertially referenced wind component

B = Y inertially referenced wind component

C = Z inertially referenced wind component

D = X body referenced wind component

E = Y body referenced wind component

F = Z body referenced wind component

It is called from subroutine WIND every INTRTE seconds.

```
Subroutine RCVANG (A,B,C,D,E,F,G,H)
  Program GDDRONE
```

This subroutine uses inertially referenced coordinates to determine azi-
muth and elevation planar angles relative to the vehicle. The arguments are:

A = X ⎫   vector coordinates to a
B = Y ⎬   point in space in iner-
C = Z ⎭   tial reference frame.
D = dummy
E = dummy
F = dummy
G = planar azimuth angle relative to vehicle boresight
H = planar elevation angle relative to vehicle boresight

It is called from subroutine RCWIND every INTRTE seconds.

D-10

```
Subroutine MICRO (A,B,C,D)
   Program GDGUID
```

This subroutine models the effects of the actual word length used in the microprocessor on which the guidance computer is implemented. The arguments are:

> A = Number of bits used to represent variables
>
> B = Scaling parameters
>
> C = The truncated quantity
>
> D = Integer representation of the truncated quantity

This subroutine is called from CRSGID every DRATE seconds.

```
Subroutine ATRPL2 (A,B,C,D,E,F,G,H)
   Program GDDRONE
```

This subroutine is a two-dimensional array lookup routine. The arguments are:

> A = Table 1 (D elements)
>
> B = Table 2 (E elements)
>
> C = Table 3 (D × E elements)
>
> F = Current A value
>
> G = Current B value
>
> H = Interpolated value

This subroutine is called form AERDYN every INTRTE seconds.

```
Subroutine BLIMIT (A,B,C)
  Program GDDRONE
```

This subroutine limits a variable between upper and lower limits. The *arguments are:*

        A = Variable to be limited

        B = Lower limit

        C = Upper limit

This subroutine is called from both EQAMO and KINEM every INTRTE seconds.

```
Subroutine INTGRT (A,B,C,D,E,F)
  Program GDDRONE
```

This subroutine is the integration procedure used in the program:

$$A = A + D(2B - C)$$

where   A is the variable being integrated

               D is the integration interval

               B is the value of A at the previous interval $(n - 1)$

               C is the value of A two intervals prior $(n - 2)$

               E = 1

               F = 1

This subroutine is called every INTRTE seconds by both EQAMO and KINEM.

```
Subroutine INVERT (A,B,C)
   Program GDDRONE
```

This subroutine transforms a vector from body to inertial coordinates. The arguments are:

> A = Boresight vector in inertial coordinates
>
> B = Vector in body coordinate frame
>
> C = Vector in space coordinate frame

This subroutine is called every INTRTE seconds from the subroutine KINEM.

```
Subroutine GYERR (A,B)
   Program GDGUID
```

This subroutine computes a bias corrupted value of the body rates to simulate gyro error. The arguments are:

> A = Uncorrupted value of the body rate as
>   input and corrupted value upon return
>
> B = Index which refers to particular value
>   of error which are stored in a common
>   array.
>   1 = yaw
>   2 = dummy
>   3 = pitch

The values for yaw and pitch bias are entered from the terminal at run time.
   This subroutine is called from EQAMO every INTRTE seconds.

REFERENCES

1. J. Capon, "Multipath Parameter Computations for the MLS Simulation Computer Program," Lincoln Laboratory, M.I.T., Project Report ATC-68 (8 April 1976). DDC-A024350/1

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-83-005	**2. GOVT ACCESSION NO.** A128 871	**3. RECIPIENT'S CATALOG NUMBER**
4. TITLE *(and Subtitle)* Description of a Digital Simulation Program for Modelling Tactical Anti-Radiation Weapons		**5. TYPE OF REPORT & PERIOD COVERED** Project Report
		6. PERFORMING ORG. REPORT NUMBER Project Report TST-69
7. AUTHOR(s) Christopher M. Sorrentino and Richard D'Amato		**8. CONTRACT OR GRANT NUMBER(s)** F19628-80-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173-0073		**10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS** Program Element Nos. 63739F/ 63601F Project Program Nos. 2355/670B
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Systems Command, USAF Andrews AFB Washington, DC 20331		**12. REPORT DATE** 28 March 1983
		13. NUMBER OF PAGES 146
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Electronic Systems Division Hanscom AFB, MA 01731		**15. SECURITY CLASS.** *(of this report)* Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

| multiple emitter simulation | terminal homing | body fixed seeker |
| anti-radiation homing | minidrone simulation | flight simulation |

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A general description is given of the computer program which simulates the performance of an anti-radiation homing weapon taking into account the properties of the seeker, assumed to be body-referenced, and the air vehicle. For the purpose of illustration, sample calculations have assumed a minidrone as the air vehicle, although the simulation has a wide range of applications to a variety of platforms. The program can simulate terminal homing in the situations where multiple, co-channel CW emitters are deployed. The three constituent simulation blocks (signal in space, seeker, and tactical vehicle) are discussed in detail, as well as models for sensor errors and wind effects.

LMED

8